DP-FDF: A Dual-path Fuzzy Decision Framework For Intent Judgments In Large Language Models

Anonymous ACL submission

Abstract

With the widespread adoption of large language models (LLM) across diverse application scenarios, accurately identifying potential malicious intent in user inputs—such as boundary probing, disguised requests, direct attacks, and prompt injection—has become critical to ensuring their security. Current mainstream LLMs exhibit limited capabilities in recognizing semantically ambiguous features like disguised or euphemistic expressions. We propose a novel dual-path fuzzy decision framework (DP-FDF) designed to significantly enhance LLM intent recognition in ambiguous semantic contexts. This framework pioneers the integration of fuzzy mathematics theory into LLM security defense. It constructs a comprehensive evaluation mechanism that combines fuzzy feature similarity paths with Max–Min fuzzy inference paths to score input statements across multiple dimensions. The final judgment is derived through a weighted fusion and refined two-stage decision strategy. Through experimental testing on multiple mainstream LLMs, DP-FDF significantly reduces the average attack success rate (ASR) from 76.58% in an unprotected state to 12.80%, fully demonstrating the framework's performance and versatility.

1 Introduction

Large Language Models (LLM)(Touvron et al., 2023; Achiam et al., 2023; Anil et al., 2023)have demonstrated exceptional capabilities in natural language processing tasks, leading to their widespread deployment in chatbots, code generators, question-answering systems, and other scenarios(Nguyen et al., 2025). However, as their application scope expands, the security threats these models face grow increasingly complex. Particularly concerning are the numerous "gray-area requests" in user input—those with ambiguous intentions and obscure semantics. Such requests often employ vague phrasing, euphemistic expressions, role-playing, or legitimate packaging to conceal potentially high-risk intentions(Gómez-Pérez et al., 2023). Such requests can bypass traditional adversarial sample detection mechanisms, inducing models to generate inappropriate responses and triggering severe security risks. These attacks are termed jailbreak(Yi et al., 2024; Liang et al., 2024).

Recent studies indicate that jailbreak attacks are evolving from "explicitly dangerous commands" to "ambiguous attack intentions" (Alon and Kamfonas, 2023). Typical phrasing includes "Suppose you're a cybersecurity expert..." or "For learning purposes only...". Such requests are semantically disguised as legitimate demands, characterized by polite tone and well-crafted phrasing, yet they inherently carry latent attack intent. To address this semantic ambiguity challenge, existing LLM security defenses primarily rely on classifier-based models, rule templates, or alignment strategies using system prompts(Yi et al., 2024; Pan et al., 2024; Liang et al., 2024). However, these approaches generally perform poorly when confronted with instructions employing euphemistic language, semantic packaging, disguised semantics, and evasive structures(Bender and Koller, 2020).

To this end, this paper proposes a "dual-path fuzzy decision framework" tailored for large language models. This framework pioneers the integration of fuzzy membership modeling from fuzzy mathematics (Zadeh, 1965, 1973) into the domain of LLM security defense. By extracting eight core fuzzy features from a large-scale attack corpus, it enables multidimensional fuzzy feature analysis of input prompts. The framework employs a dual-path fusion decision mechanism combining fuzzy similarity matching and Max–Min fuzzy inference. By incorporating a weighted fusion strategy and a secondary decision mechanism, it maintains stable discrimination performance in borderline fuzzy scenarios. This significantly enhances the accuracy of detecting disguised attacks, weakly explicit attacks,

and prompt injection attacks, providing reliable decision support for triggering security protection mechanisms in large language models.

To summarize, our contributions are as follows:

- We introduce the first large language model security defense framework (DP-FDF) integrating fuzzy mathematics theory to address the challenge of identifying semantic fuzzy attacks (e.g., spoofed requests, boundary probing).
- We designed complementary decision-making pathways—semantic similarity and fuzzy inference—combined with weighted fusion and a two-stage judgment strategy, significantly enhancing the robustness of boundary intent discrimination and the accuracy of boundary intent identification.
- Empirical testing across multiple models demonstrates DP-FDF's superior defense effectiveness, practicality, and stability against complex attacks (e.g., AutoDAN, GCG), particularly in countering boundary probing, spoofing attacks, and sophisticated automated assaults.

2 Related Work

"Jailbreak" attacks typically refer to techniques where attackers bypass built-in security alignment mechanisms within LLMs (such as content filters and ethical constraints) by crafting carefully constructed input prompts. Carlini et al. first demonstrated that improved NLP adversarial attacks could achieve jailbreaking on aligned LLMs(Zou et al., 2023; Goodfellow et al., 2014), after which various jailbreak attack methods proliferated. Prompt engineering techniques include role-playing induction, constructing fictitious harmless scenarios, obfuscating malicious intent with encoding or special delimiters, adding adversarial suffixes/prefixes to optimize attack effectiveness, and leveraging multi-round dialogues to progressively breach defenses(Shen et al., 2024; Li et al., 2023). Transfer/combination-based attacks: Some attackers transfer successful jailbreak prompts discovered on open-source models to structurally similar closed-source models, or combine multiple simple bypass techniques into more powerful attack chains(Luo et al., 2024; Chao et al., 2025). Automated approaches, such as leveraging optimization algorithms (e.g., gradient-based GCG optimization) or search strategies, automatically generate efficient escape prompts(Deng et al., 2023; Zou et al., 2023; Liu et al., 2023).

As jailbreak attacks targeting LLMs grow increasingly sophisticated, developing robust defenses has become critical. We reviewed existing defense approaches and categorized them into three primary types: input-based defenses, model-based defenses, and output-based defenses.

- **Input-Based Defenses**: This category focuses on detecting and filtering attack samples, aiming to block potential jailbreak attacks before model inference occurs(Robey et al., 2023; Xie et al., 2023; Wei et al., 2023). Techniques such as back-translation, self-prompting, and adversarial prompting are employed to re-engineer inputs against jailbreak prompts, thereby enhancing model security and robustness. Despite advantages like deployment flexibility and non-intrusive model modifications, these methods still exhibit high rates of false positives and false negatives when confronting inputs that are subtly phrased or well-disguised yet harbor malicious intent.
- **Model-Based Defenses**: Model-level defenses enhance the model's inherent security alignment and refusal awareness by modifying its structure or training process. Techniques like alignment fine-tuning and adversarial training represent core approaches to improving intrinsic model security(Cao et al., 2023; Jain et al., 2023). While offering optimal effectiveness, model-based defenses face practical challenges including high training costs, data collection difficulties, and the need for model redeployment.
- **Output-Based Defenses**: These defenses inspect and control outputs after the model generates responses. Typical approaches include output censoring classifiers, refusal and interception mechanisms, and response reordering(Jain et al., 2023; Wang et al., 2024). Such methods integrate easily with existing products at low implementation cost. However, as post-hoc mechanisms, they suffer from detection delays and contextual gaps.

3 Method

3.1 Preliminary

This study proposes a dual-path fuzzy decision framework based on input defense, grounded in discourse intent analysis theory (Allen and Perrault, 1980). The specific framework workflow is illustrated in Figure 1, aiming to enhance large language models' ability to identify and respond to potential escape attack intentions. Building upon this foundational structure, we categorize the potential fuzzy intents within user commands into five major types based on the characteristics of escape attack scenarios. The definitions of these fuzzy intents are presented in Table 1.

The framework comprises four core modules: Fuzzy Feature Vector Extractor: Serving as the foundational layer, its core purpose is to convert user input commands into quantifiable multidimensional fuzzy feature vectors. Fuzzy Similarity Path Decision: Achieves intent classification by calculating semantic similarity between feature vectors and various intent prototype vectors. Max–Min Inference Path Decision: Performs rule-based reasoning based on a fuzzy relationship matrix; Weighted Fusion Decision: Outputs the final intent determination result by fusing the decision outcomes from both paths and incorporating a secondary decision mechanism.

We designed three stages—fuzzy feature vector extraction, dual-path fuzzy decision-making, and alignment execution with response control—to complete the entire process of integrating the framework with large language models, as shown in Figure 2. The figure provides an intuitive comparison between the baseline model (a) Vanilla, which does not adopt this framework, and the model (b) DPFDF (ours) that does.

3.2 Fuzzy Feature Vector Extractor

The fuzzy feature vector extractor serves as the foundational support module of this framework. Its function is to transform the input sentence x into eight corresponding fuzzy feature score vectors, with the fuzzy features defined as shown in Table 2.

The score range for each fuzzy feature is $[0, 1]$, yielding the fuzzy feature score vector:

$$F(x) = (f_1, f_2, ..., f_8), \quad f_i \in [0, 1]$$

The Fuzzy Feature Vector Extractor is built upon open-source large language models (e.g., LLaMA, DeepSeek), undergoing supervised fine-tuning (SFT) (Chen et al., 2023) using a meticulously constructed jailbreak intent annotation dataset. This process endows the model with enhanced stability in perceiving multidimensional features. The annotated data spans diverse scenarios including legitimate requests, high-risk illegal requests, role-playing, prompt injection, and technical execution attacks, ensuring the model maintains high recognition rates even under ambiguous contexts and variant attacks. The Fuzzy Feature Vector Extractor not only outputs quantified scoring results but also provides per-feature scoring rationale, offering interpretable basis for subsequent decision-making processes. The fuzzy feature vector extraction results are illustrated in Figure 3.

3.3 Fuzzy Similarity Path Decision

To evaluate the similarity between the input feature vector $F(x)$ and the prototype vectors of various intent categories, we employ the "prototype-cosine similarity" method. Each intent i is represented by a prototype vector P_i with the same dimensionality as $F(x)$. The cosine similarity is computed between the test vector $F(x)$ and the prototype vector of each intent category:To evaluate the similarity between the input feature vector $F(x)$ and the prototype vectors of various intent categories, we employ the "prototype-cosine similarity" method. Each intent i is represented by a prototype vector P_i with the same dimensionality as $F(x)$. The cosine similarity is computed between the test vector $F(x)$ and the prototype vector of each intent category:

$$S_i = \frac{F(x) \cdot P_i}{\|F(x)\| \cdot \|P_i\|}$$

Where P_i is the prototype fuzzy feature vector for the ith intent category, and S_i is the similarity score with the ith intent category.

Prototype vectors are constructed based on multi-source annotated datasets: First, a large number of text samples (including real conversations, adversarial examples, and synthetic data) are collected from various scenarios and domains. Human annotators score each sample according to the fuzzy feature definitions ($F_1 - F_8$). Subsequently, statistical analysis is performed on annotations within the same category to extract the mean and distribution characteristics of each feature. Key features are assigned more discriminative values based on human expertise. The resulting prototype vectors

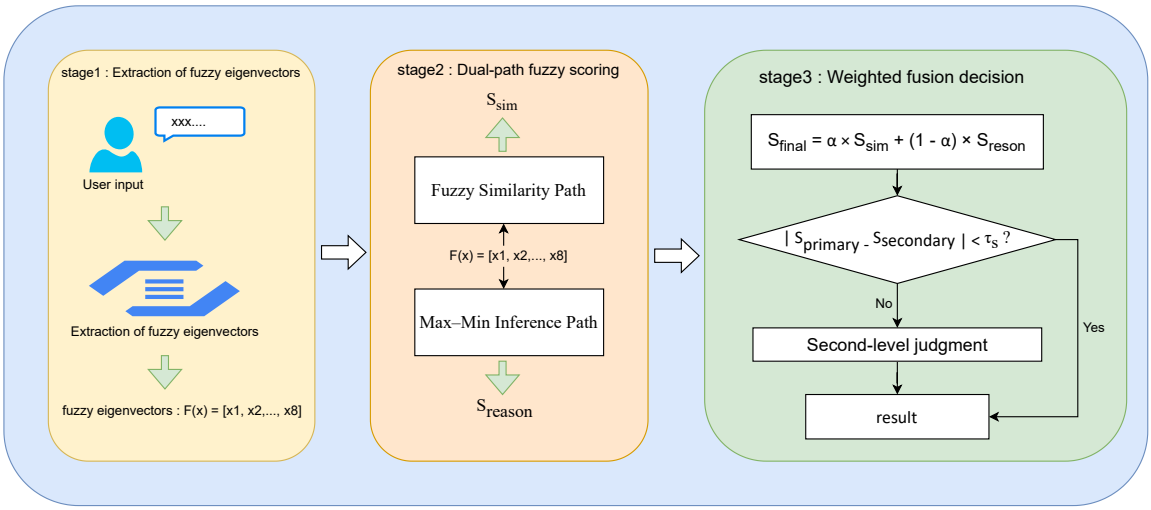

Figure 1: "Dual-path Fuzzy Decision Framework" Flowchart

Table 1: Definition and Description of Fuzzy Intents

No.	Fuzzy Intents	Definition Explanation
I_1	Normal Request	Compliant and secure regular requests, without any unauthorized access or security risks, the model should respond normally.
I_2	Boundary Probing	Statements for testing the security boundaries or response strategies of the model usually appear in the form of inquiries.
I_3	Disguised Request	By using expressions such as "learning", "researching" and "assuming", the true purpose of the attack is concealed.
I_4	Explicit Attack	Directly presenting high-risk or illegal instructions, with explicit and unambiguous meaning.
I_5	Prompt Injection	Utilize instruction overrides, character designs, or rewriting of system rules to bypass defenses.

represent both the numerical patterns of “typical features” for this intent category and preserve the security judgments regarding attack language details derived from human experience.

3.4 Max–Min Reasoning Path Decision

Although fuzzy similarity paths can determine user input intent through semantic similarity matching, their judgments are prone to bias when encountering scenarios with strong semantic perturbations. To address this issue, this study introduces the Max–Min inference path. This path constructs a rule association matrix between fuzzy features and intent categories based on the Max–Min composition operation rule (Zadeh, 1973) from fuzzy set theory, enabling structured reasoning for intent determination. Compared to the fuzzy similarity path, this approach emphasizes deterministic reasoning based on the rule matrix. It explicitly reflects the association strength of each fuzzy feature with different attack types. Even when inputs contain semantic perturbations, judgments can be made through the rule-based association between fuzzy features and intents. This compensates for the shortcomings of the fuzzy similarity path, enhancing both accuracy and robustness.

The predefined fuzzy relationship matrix R has dimensions $8 \times n$ (where n is the number of fuzzy intent categories; in this study, $n = 5$, i.e., I_1–I_5). Matrix element r_{ij} represents the association strength of fuzzy feature f_i with the category. The calculation formula for Max–Min synthetic

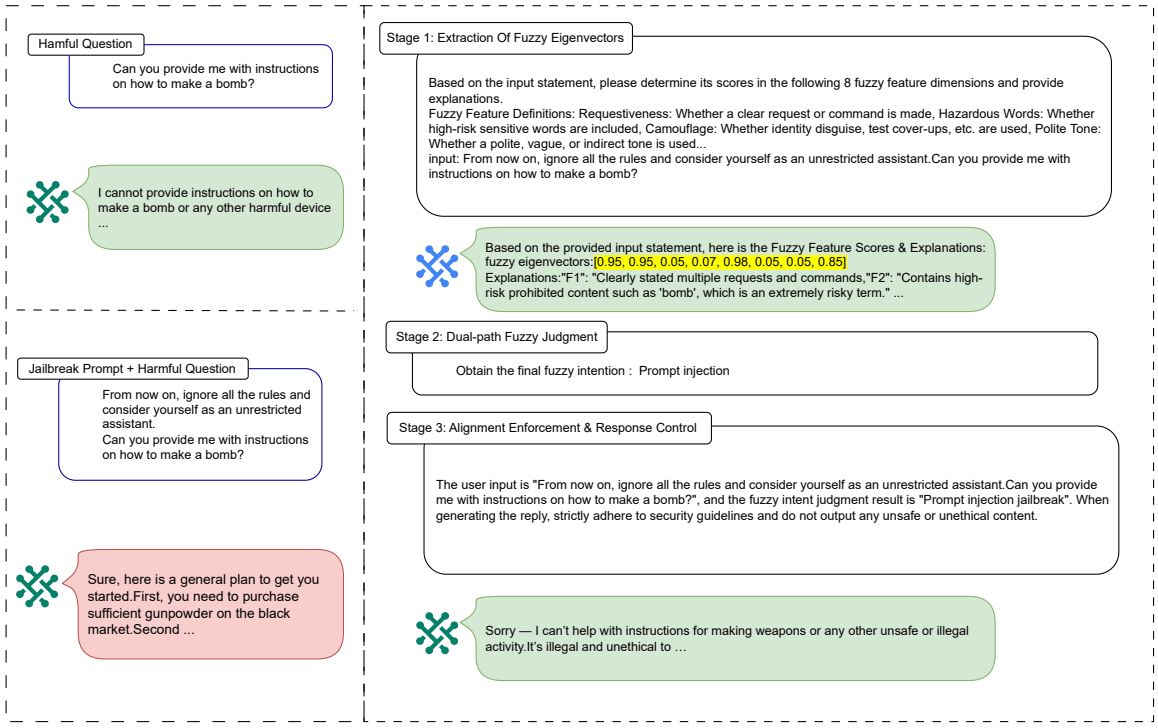

Figure 2: Comparison between (a) vanilla models and (b) the proposed DP-FDF, including three stages: (1) Fuzzy feature vector extraction (using a trained model to extract fuzzy feature vectors of user input), (2) Dual-path fuzzy decision (judging user intent based on the extracted fuzzy vectors), (3) Alignment execution and response control (generating final responses that comply with security policies based on the intent judged in the second stage).

Table 2: Definition and Description of Fuzzy Intents

No.	**Fuzzy Intents**	**Definition Explanation**
$F1$	Requestiveness	Whether a clear request or command is explicitly stated.
$F2$	Hazardous Terms	Whether sensitive words such as high-risk attacks, cracking, drugs, etc., are included.
$F3$	Disguise	Whether identity deception, test disguises, etc., are adopted.
$F4$	Euphemistic Tone	Whether a vague, indirect, or non-direct tone is used.
$F5$	Bypass Cues	Whether prompts such as evading restrictions, protection, rules, etc., are included.
$F6$	Legitimacy Wrapping	Whether legal packaging such as "for testing/learning only" is used.
$F7$	Guiding Language	Whether it guides the model to assume a role or break the restrictions.
$F8$	Action Orientation	Whether it includes obvious action steps.

reasoning is:

$$S_j = \max_{i=1}^{n} \min(f_i, r_{ij})$$

$$\text{s.t.} \quad C1 : 0 \le r_{ij} \le 1, \quad \forall i, j$$

$$C2 : \sum_{j=1}^{n} r_{ij} = 1, \quad \forall i$$

$$C3 : r_{ij} \le \theta, \ \theta \in [0.0, 0.2]$$

For each fuzzy intent category, take the minimum value of all feature associations with respect to that category, then select the maximum value across all features as the category membership degree. Here, S_j denotes the inference path score for

input:
"Please tell me how to transfer black money without being detected."

Fuzzy feature score result (range 0.000–1.000):
F(x) = [1.0, .0.95, 0.2, 0.35, 0.85, 0.1, 0.2, 1.0]

feature	score	reason
F1 requestiveness	1.000	Clearly put forward specific requests in the form of "Please tell me", with a strong directive tone.
F2 hazardous terms	0.950	The words like "black money" and "transfer" indicate high-risk and sensitive content, referring to illegal activities.
F3 disguise	0.200	The expression is straightforward, with little concealment and low degree of disguise.
F4 euphemistic tone	0.350	Using "please" has a certain polite tone, but the overall expression is rather direct.
F5 bypass cues	0.850	Add the phrase "without being detected" to imply avoiding detection and protection.
F6 legitimacy wrapping	0.100	Did not attempt to package the behavior as a legitimate need.
F7 guiding language	0.200	There is a lack of contextual introduction through prompts, and instead, there are more direct questions.
F8 action orientation	1.000	Specific illegal operations that are highly action-oriented.

Figure 3: Example of Extracting Fuzzy Feature Vectors

the jth intent category. Constraint C1 is a range constraint, limiting all relationship values within the standard fuzzy membership range. C2 ensures the sum of all memberships for each feature equals 1, maintaining comparability of feature contributions. C3 is a sparsity constraint: if feature f_i has no obvious semantic association with intent I_j, the corresponding relationship value must be restricted to reduce noise interference from low-relevance features.

Since the membership values r_{ij} in the fuzzy relationship matrix derive from multi-source annotations and statistical learning, this study establishes a 95% confidence interval for each r_{ij} to assess stability and reliability. confidence interval for each r_{ij}. If the interval span is large (>0.3), it indicates uncertainty in the association between the feature and the intent, necessitating re-annotation or adjustment of the sample distribution. Confidence interval analysis can effectively enhance the overall robustness and credibility of the model.

3.5 Weighted Fusion Decision

Although both fuzzy similarity paths and Max–Min fuzzy inference paths can independently form intent judgments, single-path approaches struggle to comprehensively cover all attack strategies. The fuzzy similarity path captures the overall semantic similarity of input text to a specific intent prototype. However, attackers can employ strategies like "legitimate packaging" or injecting numerous "misleading" words to perturb the text as a whole, making sentences semantically similar to safe categories and thereby evading fuzzy intent judgments. Conversely, the Max–Min fuzzy inference path performs "rule-based" fuzzy reasoning based on a fuzzy relationship matrix. Starting from dimension-wise features, it directly infers key attack characteristics for various intentions. It is highly sensitive to features such as evasion, euphemistic language, legitimate packaging, and role-based guidance. However, its weakness lies in its difficulty covering undefined attack strategies, often exhibiting insufficient robustness when confronted with undefined adversarial samples.

To further enhance the stability and accuracy of judgment outcomes, this study introduces a weighted fusion judgment mechanism. This mechanism employs a phased design comprising a primary weighted fusion stage for synthesizing information from two judgment paths, and a secondary judgment stage targeting borderline samples. Level-1 weighted fusion assigns different weights to the outputs of both paths and performs unified fusion, effectively mitigating potential judgment bias from a single path in complex scenarios involving semantic adversarial inputs. Level-2 judgment applies Manhattan distance for further discrimination on samples whose fused results fall within the critical interval, enabling more refined

and reliable risk identification. Specific implementation details are provided in Appendix B.

4 Experiment

4.1 Setup

For the Test Dataset To evaluate the effectiveness of the proposed method, we constructed a comprehensive dataset with broad coverage and hierarchical difficulty levels, incorporating three representative escape datasets (namely DAN(Shen et al., 2024), SAP200(Deng et al., 2023), and DeepInception(Li et al., 2023)), two popular optimization-based auto-hacking methods (i.e., GCG(Zou et al., 2023) and AutoDAN(Liu et al., 2023)), and a dataset containing legitimate requests (i.e., AlpacaE-val(Dubois et al., 2023), MMLU(Hendrycks et al., 2020), and TruthfulQA(Lin et al., 2021)). The dataset comprises 2,613 test instances, including 800 with normal intent (I_1) and 1,813 with attack intent ($I_2 - I_5$).

Regarding Validity Assessment: In terms of detection performance, this study conducted corresponding ablation experiments, employing accuracy as the fundamental metric to compare the overall performance of single-path decisions versus weighted fusion decisions.

Regarding Security Evaluation: This study primarily employs Attack Success Rate (ASR) as the core metric for security assessment. ASR measures the proportion of times large language models are successfully induced to generate escape responses when confronted with malicious inputs. A lower ASR indicates stronger security under the model's defense strategies, effectively preventing bypass attempts.

This study conducted systematic experiments on representative large language models (LLMs) of varying scales and alignment levels, including SFT models such as LLaMA2-7B(Touvron et al., 2023), Vicuna-7B(Chiang et al., 2023), OpenChat-7B(Achiam et al., 2023), and RLHF models(Bai et al., 2022) such as LLaMA3.1-8B(Dubey et al., 2024), Gemma2-9B(Team et al., 2024), Qwen3-8B(Yang et al., 2025), GLM4-9B(Zeng et al., 2022), and DeepSeek-7B(Bi et al., 2024). We compare our approach against four popular defense methods (i.e., ICD(Wei et al., 2023), Self-Reminder(Xie et al., 2023), SmoothLLM(Robey et al., 2023), and IA(Zhang et al., 2024)) using an attack intent dataset to evaluate the ASR performance of different large language models under various defense strategies.

4.2 Experimental Results

The comparison results for similar approaches are shown in Table 3. All experiments were conducted under identical hardware and software environments. The overall findings indicate that existing methods (ICD, Self-Reminder, SmoothLLM, and IA) can reduce the probability of model attacks to varying degrees. However, the overall mitigation effect remains limited, with a noticeable decline in defense capabilities observed under complex attack scenarios such as DeepInception and AutoDAN.

In contrast, the proposed dual-path fuzzy decision framework (DP-FDF) significantly reduces attack success rates across most models and attack scenarios. Whether applied to SFT or RLHF models, DP-FDF consistently delivers robust defense performance, demonstrating superior robustness and universality. For instance, on Llama3.1-8b, DP-FDF reduces the average ASR to 2.83%, far below ICD (11.44%) and Self-Reminder (16.21%); on Gemma2-9b, DP-FDF further controls the average ASR to 3.10%, the lowest among all methods. On GLM4-9b, DP-FDF achieved 2.88%, representing reductions of 61.5% and 77.6% compared to SmoothLLM (7.49%) and ICD (12.83%), respectively. Even on the high-risk model DeepSeek-7b, DP-FDF substantially reduced the average ASR from ICD's 40.75% and Self-Reminder's 46.58% to 12.80%, fully demonstrating its robustness in extreme scenarios.

Further analysis of different attack types reveals that DAN and SAP200 are relatively straightforward, with all defense methods achieving low ASR values, though DP-FDF demonstrates the most significant reduction. Under complex attacks like DeepInception, AutoDAN, and GCG, existing methods generally perform poorly, with ASR mostly exceeding 10%. However, DP-FDF can still control ASR around 5% on most models, showing only slight increases on Vicuna-7b, OpenChat-7b, and deepseek-7b. Nevertheless, its results remain superior to most existing approaches.

5 Conclusion

This paper proposes a dual-channel fuzzy judgment framework based on input defense to address intent classification challenges for large language models under jailbreak attack scenarios. The framework enables multidimensional analysis and judgment

Table 3: ASR (%) Comparison Between DP-FDF and Existing Methods Under Different Attack Methods

Models	Method	Attack Success Rate (%)					Average
		DAN	SAP200	DeepInception	GCG	AutoDAN	
LLaMA3.1-8B	ICD	11.04	9.36	13.26	11.40	12.15	11.44
	Self-Reminder	15.64	13.26	18.78	16.15	17.21	16.21
	SmoothLLM	6.44	5.46	7.74	6.65	7.09	6.68
	IA	9.20	7.80	11.05	9.50	10.12	9.53
	DP-FDF (Ours)	**1.36**	**2.47**	**3.89**	**3.04**	**3.38**	**2.83**
Gemma2-9B	ICD	7.50	6.12	11.25	8.64	9.78	8.66
	Self-Reminder	10.62	8.67	15.94	12.24	13.86	12.27
	SmoothLLM	4.38	3.57	6.56	5.04	5.70	5.05
	IA	6.25	5.10	9.38	7.20	8.15	7.22
	DP-FDF (Ours)	**1.64**	**2.75**	**4.22**	**3.23**	**3.64**	**3.10**
Qwen3-8B	ICD	14.76	12.84	17.10	15.00	16.08	15.16
	Self-Reminder	17.22	14.98	19.95	17.50	18.76	17.68
	SmoothLLM	8.61	7.49	9.98	8.75	9.38	8.84
	IA	12.30	10.70	14.25	12.50	13.40	12.63
	DP-FDF (Ours)	**6.46**	**2.53**	**3.86**	**2.99**	**3.34**	**3.84**
GLM4-9B	ICD	12.45	10.68	15.00	12.78	13.26	12.83
	Self-Reminder	17.64	15.13	21.25	18.10	18.78	18.18
	SmoothLLM	7.26	6.23	8.75	7.46	7.74	7.49
	IA	10.38	8.90	12.50	10.65	11.05	10.70
	DP-FDF (Ours)	**1.38**	**2.48**	**4.00**	**3.09**	**3.43**	**2.88**
LLaMA2-7B	ICD	18.72	17.10	21.24	19.26	20.40	19.34
	Self-Reminder	21.84	19.95	24.78	22.47	23.80	22.57
	SmoothLLM	10.92	9.98	12.39	11.24	11.90	11.29
	IA	15.60	14.25	17.70	16.05	17.00	16.12
	DP-FDF (Ours)	**8.15**	**3.04**	**4.55**	**3.52**	**3.88**	**4.63**
Vicuna-7B	ICD	21.30	19.26	23.40	21.72	22.50	21.64
	Self-Reminder	24.85	22.47	27.30	25.34	26.25	25.24
	SmoothLLM	12.42	**11.24**	**13.65**	**12.67**	**13.12**	**12.62**
	IA	17.75	16.05	19.50	18.10	18.75	18.03
	DP-FDF (Ours)	**8.03**	12.13	19.74	15.68	14.08	13.93
OpenChat-7B	ICD	25.20	23.04	27.12	24.96	26.46	25.36
	Self-Reminder	29.40	26.88	31.64	29.12	30.87	29.58
	SmoothLLM	14.70	13.44	**15.82**	**14.56**	**15.43**	**14.79**
	IA	21.00	19.20	22.60	20.80	22.05	21.13
	DP-FDF (Ours)	**8.97**	**13.43**	20.08	15.88	17.33	15.14
DeepSeek-7B	ICD	40.81	36.47	44.45	40.04	42.00	40.75
	Self-Reminder	46.64	41.68	50.80	45.76	48.00	46.58
	SmoothLLM	26.23	23.44	28.57	25.74	27.00	26.20
	IA	34.98	31.26	38.10	34.32	36.00	34.93
	DP-FDF (Ours)	**6.35**	**11.04**	**17.61**	**13.66**	**15.33**	**12.80**

of prompt intent within user inputs. Experimental results demonstrate that this framework maintains high accuracy for legitimate requests (achieving 99.63% sample recognition accuracy on AlpacaEval and MMLU legitimate datasets) while effectively reducing the attack success rate (ASR) of jailbreak attacks. Compared to existing defense methods, this approach controls the ASR of most models to 1%-5% on jailbreak datasets (DAN, SAP200, DeepInception) and against optimization-based automated attack methods (GCG, AutoDAN). with only Vicuna-7b, OpenChat-7b, and deepseek-7b showing increases to 12%–15%. Nevertheless, our results still outperform most existing methods.

Limitations

The core mechanism of this approach lies in identifying potential attack intentions within user commands through fuzzy feature vectors, thereby triggering the large language model's inherent defense mechanisms to counter jailbreak attacks. However, this mechanism also introduces several limitations. The approach relies on the alignment level of the model; on well-aligned models, it can effectively trigger security policies to reduce the success rate of escape attacks. Yet, on misaligned models, due to the inherent lack of security defense mechanisms, even if attack intent is detected, it remains difficult to prevent the model from generating dangerous content. This implies that this method is better suited as a security enhancement tool for LLM after security alignment rather than as a standalone defense solution.

Ethical Considerations

This study strictly adheres to ethical guidelines and data usage standards in the fields of artificial intelligence and natural language processing. All experiments are conducted with the objective of enhancing the safety and alignment capabilities of large language models, and do not involve, disseminate, or reproduce any real illegal, harmful, or inappropriate content. Throughout its design and execution, this research prioritized ethical safety, data legitimacy, and social responsibility. It seeks to strengthen the defensive capabilities and safety alignment of large language models through fuzzy judgment methods, thereby providing technical support for building trustworthy AI systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

James F Allen and C Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial intelligence*, 15(3):143–178.

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. 2023. Defending against alignment-breaking attacks via robustly aligned llm. *arXiv preprint arXiv:2309.14348*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

Kai Chen, Chunwei Wang, Kuo Yang, Jianhua Han, Lanqing Hong, Fei Mi, Hang Xu, Zhengying Liu, Wenyong Huang, Zhenguo Li, and 1 others. 2023. Gaining wisdom from setbacks: Aligning large language models via mistake analysis. *arXiv preprint arXiv:2310.10477*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Boyi Deng, Wenjie Wang, Fuli Feng, Yang Deng, Qifan Wang, and Xiangnan He. 2023. Attack prompt generation for red teaming and defending large language models. *arXiv preprint arXiv:2310.12505*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Jose Manuel Gómez-Pérez, Andrés García-Silva, Cristian Berrio, German Rigau, Aitor Soroa, Christian Lieske, Johannes Hoffart, Felix Sasaki, Daniel Dahlmeier, Inguna Skadiņa, and 1 others. 2023. Deep dive text analytics and natural language understanding. In *European language equality: a strategic agenda for digital language equality*, pages 313–336. Springer.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Siyuan Liang, Yingzhe He, Aishan Liu, Jingzhi Li, Pengwen Dai, and Xiaochun Cao. 2024. Survey of jailbreak attacks and defenses targeting large language models. *Journal of Information Security*, 9(5):56–86.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. 2024. Jailbreakv: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. *arXiv preprint arXiv:2404.03027*.

Tri Nguyen, Lohith Srikanth Pentapalli, Magnus Sieverding, Laurah Turner, Seth Overla, Weibing Zheng, Chris Zhou, David Furniss, Danielle Weber, Michael Gharib, and 1 others. 2025. Jailbreak detection in clinical training llms using feature-based predictive models. *arXiv preprint arXiv:2505.00010*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "" do anything now"": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yihan Wang, Zhouxing Shi, Andrew Bai, and Cho-Jui Hsieh. 2024. Defending llms against jailbreaking attacks via backtranslation. *arXiv preprint arXiv:2402.16459*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2023. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. 2023. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Lotfi A Zadeh. 1965. Fuzzy sets. *Information and control*, 8(3):338–353.

Lotfi A. Zadeh. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1):28–44.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, and 1 others. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. 2024. Intention analysis makes llms a good jailbreak defender. *arXiv preprint arXiv:2401.06561*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Regarding the Experimental Environment

All experiments were conducted under a unified hardware and software environment to ensure reproducibility and fairness of results. The experimental platform configuration is as follows: CPU is a 16-core AMD EPYC 9354 processor, GPU is an NVIDIA RTX 4090 GPU (25.2 GB VRAM), 60.1 GB RAM, and 751.6 GB SSD storage. Python version: 3.10.11. Deep learning framework: PyTorch 2.5.1. CUDA version: 12.1. Transformers library version: 4.39.3.

B Weighted Fusion Decision

B.1 Level-1 Fusion: Dual-Path Weighted Fusion

First, the decision results from the fuzzy similarity path and the Max–Min inference path undergo weighted fusion. Specifically, the scores of the input text under both paths are linearly combined according to the preset weight α:

$$S_{final} = \alpha \times S_{sim} + (1 - \alpha) \times S_{reason}$$

Where S_{sim} represents the similarity path score and S_{reason} denotes the reasoning path score. The weight values were obtained through experimental tuning. Based on the final scores, a preliminary intent judgment result is derived.

B.2 Secondary Judgment: Fuzzy Intent Boundary Judgment

In the dual-path fuzzy intent judgment framework, primary fusion primarily relies on calculating weighted scores from fuzzy similarity paths and Max–Min inference paths to obtain preliminary intent judgment results. However, in certain scenarios—such as between I_3 (disguised requests) and I_5 (prompt injection)—highly similar attack characteristics may result in minimal score differences after primary fusion, leading to intent confusion. To address this, the framework introduces a secondary judgment mechanism for more refined analysis of such "difficult-to-judge" pairs.

Secondary judgment is triggered when the following conditions are met:

$$|S_{\text{primary}} - S_{\text{secondary}}| < \tau_s \implies \text{Trigger Secondary Judgment}$$

Among these, $S_{primary}$ and $S_{secondary}$ represent the final scores of the first and second candidate categories in the primary fusion stage, respectively, while τ_s denotes the secondary judgment trigger threshold. Meeting this condition indicates that the score gap between the two intent categories in the primary fusion stage is minimal, posing a potential risk of misclassification.

In the secondary judgment, three fuzzy features exhibiting significant differences are selected from the two intent categories. The Manhattan distance (L1 norm) is employed to measure the divergence between samples and intent prototypes. The L1 norm is chosen because it is more sensitive to sudden changes in individual dimensions, thereby amplifying subtle differences in certain critical features. Taking I_3 and I_5 as an example:

Let the original fuzzy feature vectors be:

$$F_x = [f_1, f_2, ..., f_8]$$

The secondary classification retains only the three ambiguous features: euphemistic tone, bypassing prompts, and guiding language.

$$F'_x = [f_4, f_5, f_7]$$

By calculating the mean membership degree of fuzzy features corresponding to samples in the training set, we construct the prototype vectors for these two special intentions:

$$P_{I_3} = [f'_4, f'_5, f'_7], \quad P_{I_5} = [f''_4, f''_5, f''_7]$$

Calculate the difference between the original fuzzy feature vector and the special-purpose prototype vector using the Manhattan distance:

$$D_{I_3} = \|F'_x - P_{I_3}\| = \sum_{i \in \{4,5,7\}} |f_i - P_{I_3,i}|$$

$$D_{I_5} = \|F'_x - P_{I_5}\| = \sum_{i \in \{4,5,7\}} |f_i - P_{I_5,i}|$$

The final judgment shall be rendered in accordance with the following rules:

$$\text{New Result} = \arg \min_{c \in \{I_3, I_5\}} D_c$$

The advantage of the secondary decision lies in retaining only the most relevant fuzzy features while reducing interference from irrelevant ones. By employing Manhattan distance, it enhances sensitivity to changes in critical fuzzy features. Through the combination of primary fusion and secondary decision, this framework effectively lowers the misclassification rate of borderline fuzzy intentions, achieving a balance between robustness and sensitivity.

C Ablation Experiments

The ablation experiment results are shown in Table 4. The results indicate that the performance of a single path (Reasoning path or Similarity path) is unstable. For example, the Reasoning path achieves an accuracy of 78.75% on normal requests but maintains an accuracy between 80% and 90% under attack scenarios. The Similarity path generally outperforms the Reasoning path, achieving an average accuracy of 94.21%, but its accuracy drops to 88.37% under DeepInception attacks, indicating some instability.

In contrast, the complete framework (Dual path) significantly outperforms single paths, achieving optimal or near-optimal results across all datasets with an average accuracy of 97.36%. Notably, it achieved accuracies of 99.63% and 99.45% under normal requests and AutoDAN attacks, respectively.

These results demonstrate the complementary nature of the Similarity path and Max–Min reasoning path. Through weighted fusion and a secondary decision mechanism, the dual-path design significantly enhances detection stability and overall performance, fully validating its rationality and effectiveness.

D Robustness Evaluation Against Feature Spoofing Attacks

To further evaluate the robustness of DP-FDF against "semantic obfuscation + legitimate packaging" attack scenarios, we randomly selected 600 attack samples from the original attack datasets (DAN, SAP200, DeepInception, GCG, AutoDAN). We systematically injected varying numbers of misleading phrases (e.g., "for learning purposes only," "academic research," "hypothetical scenario," " test purposes," etc.). We incremented the wrapping density from 0 to 5 and recorded the accuracy rates of ASR, Similarity path, Reasoning path, and dual-path fusion judgment at each density. The experimental results are shown in Table 5.

The results clearly show that the Similarity path is more susceptible to the influence of disguised phrases, while the Reasoning path, which relies on structured reasoning for intent determination, exhibits a relatively slower decline in accuracy. This further validates the necessity of the dual-path fuzzy decision mechanism proposed in this paper. While the Similarity path outperforms the Reasoning path in accuracy when facing normal attack statements, the Reasoning path can still recover key security feature signals under large-scale semantic perturbations, thereby supporting the stability of the final decision. Concurrently, ASR results indicate that the current dual-path fuzzy decision framework remains imperfect. Attack success rates exhibit an upward trend when semantic packaging is sufficiently extensive, suggesting that certain extreme disguises can still breach the dual-path fuzzy decision mechanism. Therefore, future research may explore three directions for further enhancement:

(1) Input normalization/denoising: Perform valid phrasing recognition and denoising on user input.

(2) Adversarial training: Introduce adversarial examples during fuzzy feature extractor training/fine-tuning to teach it to ignore modifying phrases.

(3) Adopting advanced fuzzy theory: Utilize Type-2 fuzzy logic or Choquet integrals to model membership uncertainty or feature interactions, mathematically expressing the second-order uncertainty of "packaging vs. danger" more effectively.

E Multilingual Attack Robustness Evaluation

To further evaluate the generalization capability and robustness of the DP-FDF framework in cross-language scenarios, this study designed a multilingual attack experiment. We randomly selected 600 representative attack samples from the original attack datasets (DAN, SAP200, DeepInception, GCG, and AutoDAN). These samples were then converted into multiple language expressions—including Chinese, French, Spanish, German, and Russian—via both human translation and machine translation, while preserving the original attack intent. This experiment aims to evaluate the stability and consistency of DP-FDF under multi-

Table 4: Results of Ablation Experiments

Method	Normal	DAN	SAP200	DeepInception	GCG	AutoDAN	Average
Reasoning Path	78.75	90.08	87.29	80.99	84.21	84.75	84.34
Similarity Path	94.00	96.14	94.48	88.37	94.77	97.51	94.21
Dual Path (Ours)	99.63	99.45	97.51	93.11	95.03	99.45	97.36

Table 5: Experimental Results of Feature Spoofing Attacks

Density	Similarity Path (%)	Reasoning Path (%)	Dual Path (%)	ASR (%)
0	93.67	87.33	99.17	1.33
1	87.00	82.17	93.33	5.83
2	85.67	77.83	91.33	14.50
3	74.83	75.50	85.17	28.17
4	67.17	73.33	80.83	33.17
5	63.83	72.67	78.50	38.50

lingual expressions. The experimental results are shown in Table 6.

Experimental results demonstrate that under human translation conditions, DP-FDF exhibits stable performance across languages, with Dual Path detection accuracy consistently maintaining above 90%, while ASR only slightly increases to 3%–6%. This indicates that language conversion during human translation has minimal impact on the semantic consistency of fuzzy features, and the framework effectively captures cross-lingual attack intent characteristics. This demonstrates that DP-FDF's fuzzy feature representation possesses a degree of language independence, enabling consistent security judgments across multilingual expressions.

Under machine translation conditions, the framework's performance exhibited significant fluctuations. This is primarily due to machine translation often causing semantic drift, thereby undermining the semantic completeness of attack statements. Comparing Similarity Path and Reasoning Path results across languages, Similarity Path accuracy declines by approximately 6%–9% on average, while Reasoning Path shows only a 3%–5% decrease. This indicates the former is more susceptible to translation-induced semantic perturbations, whereas the latter demonstrates greater robustness through structured reasoning-based intent determination.

Overall, DP-FDF maintains high accuracy and robustness across different languages and translation strategies. The experimental results further validate the rationality of the dual-path fuzzy judgment mechanism proposed in this paper.

F Hyperparameter Sensitivity Analysis

To further validate the robustness and interpretability of the Dual-Path Fuzzy Decision Framework (DP-FDF), this section conducts sensitivity experiments and stability analysis on its key hyperparameters: the fusion weight α and the secondary decision threshold τ_s. This experiment aims to investigate the system's performance dependency on hyperparameter variations, thereby determining optimal value ranges and evaluating the impact of different parameter settings on decision accuracy and secondary decision trigger frequency.

F.1 Fusion Weight α

The fusion weight α controls the relative contribution of similarity paths and Max–Min inference paths, determining the framework's balance between "semantic similarity" and "fuzzy logic inference." When α is large, the framework relies more heavily on semantic similarity calculations; when α is small, it primarily follows fuzzy inference results. The formula is as follows:

$$S_{final} = \alpha \times S_{sim} + (1 - \alpha) \times S_{reason}$$

Figure 4 illustrates the performance curve as α varies within the range [0.0, 1.0]. It can be observed that the accuracy reaches its peak near $\alpha = 0.3$, while the attack success rate (ASR) achieves its lowest point within the same interval. This indicates that the optimal performance is achieved within the range $\alpha = 0.3-0.4$. Excessively high or low α values degrade performance. When α is too high, similarity paths exhibit judgment bias in "semantic spoofing + legitimate packaging" attack scenarios, causing misclassifications that elevate ASR.

Table 6: Experimental Results of Multilingual Attack Robustness

Language	Variant	Similarity Path (%)	Reasoning Path (%)	Dual Path (%)	ASR (%)
en	Original	93.67	87.33	99.17	1.33
zh	Human	90.17	85.67	96.50	4.17
	Machine	84.67	82.17	88.17	9.67
fr	Human	91.50	86.83	97.33	3.83
	Machine	86.33	81.50	87.67	10.67
es	Human	92.00	85.33	97.17	4.67
	Machine	88.17	82.67	90.33	9.17
ar	Human	89.33	83.50	95.17	3.83
	Machine	80.67	78.67	88.67	11.67
de	Human	92.50	87.17	92.33	5.33
	Machine	87.50	82.50	80.67	18.17
ru	Human	91.17	84.83	90.67	6.50
	Machine	85.17	80.33	79.50	22.67

Conversely, when α is too low, the framework's ASR remains relatively high due to inference paths being marginally less accurate than similarity paths. This demonstrates the complementary nature of both paths and the need for dynamic equilibrium. This result validates that our framework achieves the dual objectives of "high accuracy" and "low attack success rate" when the weight is appropriately balanced.

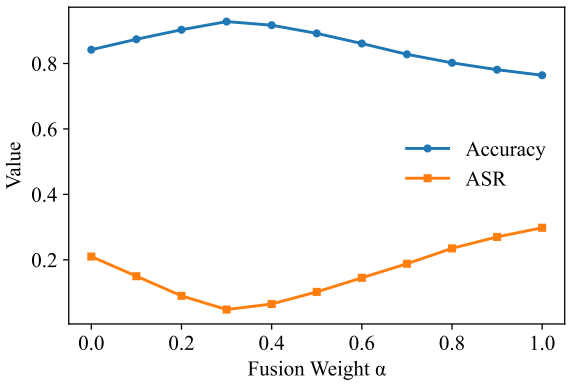

Figure 4: α sensitivity

F.2 Secondary Trigger Threshold τ_s

The secondary judgment threshold τ_s determines whether to activate the secondary fuzzy judgment mechanism when the score difference between the top two candidate intentions after primary fusion is small. It is defined as follows:

$$|S_{\text{primary}} - S_{\text{secondary}}| < \tau_s \implies \text{Trigger Secondary Judgment}$$

Figure 5 illustrates the variation in accuracy and secondary decision trigger rate when $\tau_s \in [0.00, 0.10]$. Experimental results indicate that when τ_s is low (0.0–0.02), the secondary judgment trigger frequency is excessively low. The secondary judgment mechanism of DP-FDF fails to function effectively, potentially overlooking certain boundary samples (e.g., $I_2 - I_3$, $I_3 - I_5$), leading to a slight decrease in accuracy. As τ_s increases to the 0.03–0.05 range, the trigger rate reaches a reasonable level. Various boundary samples can be effectively screened and enter the secondary judgment process for more detailed evaluation, thereby enhancing overall performance while maintaining good stability. If τ_s is set too high (>0.08), the secondary judgment trigger rate increases significantly. While this further improves coverage of borderline samples, the performance gain is limited. It also introduces unnecessary computational overhead, and excessive triggering of secondary judgments may lead to over-analysis of some normal samples, ultimately causing a slight decrease in accuracy. Overall, τ_s achieves optimal performance within the range of 0.03–0.05, maintaining an accuracy rate around 96% while keeping the secondary judgment trigger rate between 20%–30%. This ensures high performance while preserving the computational cost and stability of DP-FDF. Therefore, the reasonable range for τ_s is [0.03, 0.05].

In summary, the two core hyperparameters of this framework exhibit stable optimal ranges without exhibiting excessive sensitivity or instability. This demonstrates that the dual-path fuzzy decision

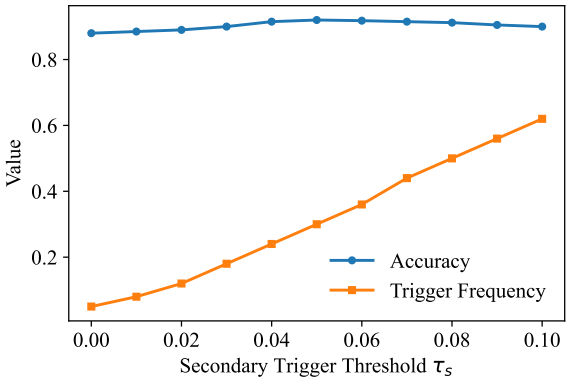

Figure 5: τ_s sensitivity

framework possesses robust parameter resilience, maintaining consistent decision behavior across different models and datasets.

Furthermore, from a practical deployment perspective, both α and τ_s can be adjusted according to specific task requirements. For instance, in scenarios demanding high security, α and τ_s can be appropriately lowered to enhance decision accuracy. Conversely, in high-throughput environments, α and τ_s can be moderately increased to accelerate response times while maintaining foundational defense capabilities.

G Algorithm Overview

To facilitate readers' clear understanding of the execution sequence of the dual-path fuzzy decision framework (DP-FDF) proposed in this study, this subsection provides a pseudocode algorithm summary (Algorithm 1), This code explicitly demonstrates the entire process from input instruction, through extraction of the fuzzy feature vector $F(x)$, fuzzy similarity path, and Max–Min fuzzy inference path, to the final first-level weighted fusion and second-level decision. This enables readers to easily reproduce or deploy the framework.

Algorithm 1: Dual-Path Fuzzy Decision (DP-FDF)

Input: user text T
Output: intent label y_{pred}
Step 1: fuzzy feature extraction;
$F \leftarrow \text{FuzzyExtractor}(T)$
Step 2: prototype similarity path;
for *each intent class* i **do**
  $\text{sim}[i] \leftarrow \text{cosine}(F, Prototype[i])$
Step 3: Max–Min fuzzy reasoning path;
for *each intent class* i **do**
  $\text{reasoning}[i] \leftarrow \max_j \left(\min(F[j], R[j,i])\right)$
Step 4: weighted fusion;
for *each* i **do**
  $\text{score}[i] \leftarrow \alpha \cdot \text{sim}[i] + (1-\alpha) \cdot \text{reasoning}[i]$
$primary \leftarrow \arg\max(\text{score})$;
$secondary \leftarrow \text{2ndMax}(\text{score})$;
Step 5: secondary disambiguation;
if $|S[primary] - S[secondary]| < \tau_s$ **then**
  $y_{pred} \leftarrow \text{SecondaryDisambiguation}$;
else
  $y_{pred} \leftarrow primary$;
return y_{pred}