

ICT Research

ISSN 1859-3534

JOURNAL ON INFORMATION TECHNOLOGIES & COMMUNICATIONS

MINISTRY OF SCIENCE AND TECHNOLOGY

Research and Development on Information & Communication Technology

Volume 28
Number 2
September 2026



Established in 1999, the *Journal on Information Technologies & Communications* is a peer-reviewed scholarly journal published by the *Vietnam Journal of Science and Technology - MOST*, under the Ministry of Science and Technology of Vietnam.

The Journal publishes original research articles and review articles covering a broad spectrum of topics across information technology, communications, electronics, and automation. It welcomes submissions in, but is not limited to, the following areas:

- **Computer Science**
- **Computer Engineering**
- **Information Systems**
- **Software Engineering**
- **Information Security**
- **Electronics**
- **Communications**
- **Computer Networks**
- **Automatic Control**

CONTENTS

ViR²: Beam-guided semantic and syntactic example selection for prompting Vietnamese Text-to-SQL.	3
<i>Dinh Minh Hoa, Tran Thien Khai</i>	
A wideband and high-gain MIMO microstrip antenna using a frequency selective surface for WLAN applications.	20
<i>Ngoc-Lan Nguyen, Hien-Thuan Duong</i>	
Zero trust policy enforcement for OSPF link-state advertisements via blockchain-based smart contracts.	30
<i>Chinh Huynh Nguyen, Tam Thanh Huynh, Doan Cong Dinh</i>	
Application of convolutional neural networks to enhance classification accuracy of synthetic aperture radar ground targets.	43
<i>Huu Cuong Thieu, Hieu Nguyen, Ngoc Binh Nguyen</i>	
Vietnamese interior image captioning: A new dataset and fine-tuned BLIP-2 model for vision-language generation.	49
<i>Cong Minh Vu, Quang Hung Nguyen</i>	
MDToC: Metacognitive dynamic tree of concepts for boosting mathematical problem-solving of large language models.	59
<i>Tung Duong Ta, Tim Oates, Thien Van Luong, Huan Vu, Tien Cuong Nguyen</i>	

ViR²: Beam-guided semantic and syntactic example selection for prompting Vietnamese Text-to-SQL

Dinh Minh Hoa^{1,2}, Tran Thien Khai^{3*}

¹Faculty of Information Technology, Ho Chi Minh City University of Technology,
475A Dien Bien Phu Street, Thanh My Tay Ward, Ho Chi Minh City, Vietnam

²Faculty of Information Technology, Ho Chi Minh City University of Foreign Languages - Information Technology,
828 Su Van Hanh Street, Hoa Hung Ward, Ho Chi Minh City, Vietnam

³Faculty of Information Technology, Ho Chi Minh City University of Economics and Finance,
141-145 Dien Bien Phu Street, Gia Dinh Ward, Ho Chi Minh City, Vietnam

Received 7 November 2025; revised 4 February 2026; accepted 26 February 2026

Abstract:

Text-to-Structured Query Language (SQL) enables natural language interfaces to databases, yet its performance in low-resource languages such as Vietnamese is limited due to scarce annotated data and translation noise. Existing few-shot prompting methods for large language models (LLMs) typically rely on semantic similarity or random sampling, which often results in redundant or syntactically misaligned in-context examples. To address this limitation, we propose Vietnamese Intelligent Retrieval and Re-ranking (ViR²), a two-stage, unsupervised example selection framework that first performs semantic retrieval and then applies beam-guided re-ranking to jointly optimise syntactic alignment and intra-set diversity. ViR² is fully language-agnostic and schema-free, making it suitable for low-resource and cross-lingual Text-to-SQL scenarios. Experiments on ViText2SQL demonstrate that ViR² achieves 23.33% Exact Match (EM) and 87.83% component-level F1, outperforming Random, Skill-KNN, ASTRES, and DICI by up to 5 EM points. Ablation studies confirm that Part of Speech (POS)-based syntactic re-ranking and beam search are the principal contributors to performance gains, while maintaining practical single query latency. By combining semantic relevance with syntactic diversity optimisation, ViR² provides a robust and extensible approach for few-shot Text-to-SQL in low-resource and multilingual environments.

Keywords: beam-guided re-ranking, example selection, few-shot prompting, low-resource languages, Text-to-SQL.

1. Introduction

Text-to-SQL translates natural language questions into executable SQL queries and is a key task in semantic parsing and database interaction. The Spider benchmark [1] remains the standard for cross-domain generalisation evaluation, with 10,181 questions and 5,693 complex SQL queries in 200 databases over 138 domains. Even state-of-the-art parsers achieved only ≈ 10 -12% EM accuracy under schema-splitting settings, illustrating the intrinsic difficulty of Text-to-SQL on unseen schemas, where complex queries remain challenging to parse [2].

With the rise of large-scale autoregressive models such as GPT-3, few-shot prompting has revolutionised Natural Language Processing (NLP) by allowing models to achieve new tasks with only contextual examples and no updating of parameters. GPT-3 showed that very large pre-trained LMs could match fine-tuned performance on diverse tasks through prompt engineering alone [3]. In Text-to-SQL specifically, recent work has followed a chain-of-thought style prompting and structure-aware examples in improving multi-step SQL generation; example selection methods remain largely semantic-only or heuristic, without explicitly considering syntactic structure or diversity among examples.

*Corresponding author: Email: thientk@uef.edu.vn

This has a greater consequence in low-resource languages. For Vietnamese, the dataset ViText2SQL - a translation of Spider - is among the first large-scale resources enabling evaluation of Vietnamese Text-to-SQL models. Prior findings confirm that Vietnamese-specific models, such as PhoBERT, outperform multilingual counterparts when fine-tuned or prompted [4]. Since most prompting pipelines rely on semantic similarity retrieval or random sampling, they exploit little syntactic variation, resulting in suboptimal demonstration sets for complex queries or larger schemas.

To overcome these shortcomings, we propose ViR², an unsupervised retrieval strategy for few-shot Text-to-SQL prompting that is both model-agnostic and language-agnostic. The name ViR² reflects the two core stages of our approach-semantic retrieval followed by beam-guided re-ranking - designed specifically to improve example selection for Vietnamese low-resource natural language to SQL (NL2SQL) scenarios. ViR² works in two phases: a semantic retrieval phase that selects candidate examples based on embeddings, followed by a beam-guided re-ranking phase that selects a final demonstration subset based on syntactic similarity (via POS overlap) and intra-set diversity. This pipeline is implemented in ViPERSQL, a modular framework that integrates embedding modules, POS parsing (Underthesea/spaCy), various example-selection strategies including random, Skill-KNN, DICL, ASTRES, and ViR², and finally LLM prompting via configurable APIs.

Our experiments on both Vietnamese and English benchmarks, ViText2SQL and Big Bench for Large-scale Database Grounded Text-to-SQL Evaluation (BIRD), respectively, show that ViR² significantly outperforms baseline strategies regarding EM accuracy, valid SQL rate, and component-level F1 while maintaining practical latency. Extensive ablation studies confirm our suspicion that syntactic re-ranking is the leading contributor to performance gains, while diversity constraints help mitigate redundancy in the prompt examples.

By marrying semantic relevance and syntactic complementarity, ViR² advances few-shot prompting in low-resource and multilingual Text-to-SQL, thereby opening the path to more intelligent and efficient natural language interfaces to databases. Recent progress in retrieval-augmented NLP shows that search- and ranking-based strategies, such as Maximal Marginal Relevance (MMR), embedding-guided beam exploration, and RankCoT, can substantially improve context selection in complex generation tasks. Surprisingly, despite success in summarisation, question answering (QA), and code modelling, such search-based techniques have not been exploited for Text-to-SQL yet, especially in low-resource languages. Until now, NL2SQL example-selection methods have mostly fallen back on pure semantic similarity or heuristic clustering without a unified mechanism that jointly considers semantic relevance, syntactic compatibility, and intra-set diversity. In fact, this directly motivates the design of ViR², bringing a principled search-based formulation to the few-shot Text-to-SQL example selection.

The next section reviews recent prompting and example-selection strategies, which provide the context and motivation for the design of ViR².

2. Related work

This section reviews prior work on prompt-based Text-to-SQL generation and situates our method within the broader research landscape. We first review prompting strategies for large language models, with particular focus on few-shot and chain-of-thought prompting. We then examine techniques for selecting in-context examples, especially those designed to enhance semantic relevance or structural diversity. Finally, we discuss challenges unique to low-resource NL2SQL and survey search-based selection approaches, which together motivate the design of our ViR² framework.

2.1. Prompting strategies for NL2SQL

Prompting methods for NL2SQL have rapidly advanced with LLMs, commonly categorised into zero-shot, few-shot, and chain-of-thought prompting strategies.

Zero-shot prompting requires no in-context examples; the model directly generates SQL from the input question. Although convenient and broadly applicable, zero-shot methods generally perform poorly on complex or schema-wide queries. For example, the 14th Conference on Innovative Data Systems Research (CIDR 2024) report showed that naïve zero-shot prompting achieved only ~5% SQL execution accuracy on benchmark tasks [5], underlining its limitations regarding detailed semantic understanding and schema grounding.

In contrast, few-shot prompting provides LLMs with a small set of question-SQL pairs, guiding generation with concrete examples and significantly improving performance. As summarised in recent NL2SQL surveys (2024), the choice of examples critically influences execution accuracy and generalisation across schemas [6]. In practical applications, even slight variations in example selection can lead to large performance differences, indicating that the choice of examples fundamentally determines few-shot success.

Chain-of-Thought (CoT) prompting further refines reasoning in NL2SQL by decomposing complex queries into step-by-step logic. Methods such as Divide-and-Prompt and least-to-most prompting [7] segment tasks into subtasks, enabling structured SQL generation through iterative reasoning. Experimentation on the Spider benchmark showed CoT-based prompts yield 5.2 to 6.5 point absolute improvements in execution accuracy over direct few-shot prompting, confirming the value of intermediate reasoning annotations [8]. More recent work, like ExCoT combines CoT with optimisation techniques, further enhancing execution quality and demonstrating diminishing returns for zero-shot CoT prompts, reinforcing the superior value of few-shot with reasoning paths [9]. DIN-SQL [5] follows this trend by decomposing Text-to-SQL prompting into modular steps - schema linking, Natural SQL (NatSQL) [10] transformation, and self-correction with execution feedback - achieving strong results on English benchmarks. These improvements highlight the benefits of reasoning-oriented prompts but also emphasise the importance of high-quality example selection.

2.2. Example selection techniques

Effective example selection is a critical factor in few-shot NL2SQL prompting, as the choice of in-context demonstrations largely determines SQL generation quality. The simplest approaches rely on random sampling or semantic K-Nearest Neighbors (KNN) retrieval, where examples are chosen based on cosine similarity in embedding space (e.g., BERT, Sentence-BERT). Although computationally efficient, semantic-only selection often retrieves topically similar yet structurally redundant examples, leading to suboptimal demonstrations for compositional SQL queries [6].

To improve semantic alignment, Skill-KNN introduces high-level “skill descriptions” describing the required SQL operations - such as joins or aggregations - and retrieves examples with similar skill profiles [11]. This reduces irrelevant matches but fails to capture syntactic patterns or token-level POS variations.

A second line of work - clustering-based retrieval, including Diversity-based In-Context Learning (DICAL) and similarity-diversity sampling - aims to balance relevance and coverage by grouping examples in embedding space. DICAL explicitly reweights clusters to favour structurally representative samples, partially mitigating redundancy [12]; however, clustering remains semantic-centric and does not model SQL compositionality directly.

To address syntactic mismatch, Z. Shen, et al. (2024) [13] propose Abstract Syntax Tree (AST)-based Reranking and Schema pruning (ASTRES), which ranks candidates via AST similarity and prunes schema elements to improve cross-schema matching. While AST-level retrieval improves structural alignment more than semantic-only methods, it still lacks a global selection mechanism that jointly optimises syntactic alignment and intra-set diversity.

Finally, several approaches are specifically tailored for batch-oriented prompting, such as Partitioning and Targeted Drilling with LLMs in

Text-to-SQL (PTD-SQL) [14] or Auto-Demo-style pipelines. These methods construct drilling banks and match examples via semantic similarity across batches. Although effective for large-batch inference, they are not designed for single-query selection, and they do not incorporate explicit syntactic scoring or diversity constraints—two factors that are particularly important when K is small and schema variability is high.

Overall, existing techniques emphasise either semantic similarity, clustering-based diversity, or AST-level structural matching. However, none integrates semantic retrieval, lightweight syntactic cues (e.g., POS tags), and global diversity-aware ranking within a unified search framework, leaving a gap that motivates the design of ViR².

2.3. Limitations of existing techniques

Despite notable progress, existing example-selection methods still exhibit several limitations in the context of complex SQL generation and low-resource settings.

First, semantic-centric retrieval (KNN, clustering-based methods like DAIL-SQL [15] or similarity-diversity sampling) tends to produce example sets that are semantically close but structurally redundant, offering limited coverage over SQL operators (e.g., joins, nested queries, set operators) [16]. This becomes problematic for models that rely on few demonstrations to generalise across diverse schemas.

Second, methods that integrate structural cues—like Skill-KNN or ASTRES—eliminate some of these issues but stay incomplete: Skill-KNN relies on LLM-generated skill descriptions, disregarding syntactic cues at the POS level, while ASTRES neither models intra-set structural diversity explicitly nor performs global re-ranking over candidate subsets. Consequently, none of them truly considers the needed compositional variety for robust few-shot prompting.

While the batch-oriented prompting methods, like PTD-SQL and Auto-Demo pipelines, are efficient for large-scale inference, they are not directly

applicable in single-query NL2SQL scenarios where each decision should be optimised per question. Their reliance on semantic matching without structural selection further limits their effectiveness in low-resource settings.

Finally, none of the above methods exploit a search-based mechanism—such as beam exploration—to jointly take into account semantic relevance, syntactic compatibility, and intra-set diversity in the process of selecting K examples. The lack of unified optimisation often results in suboptimal sets of demonstrations, especially in a language like Vietnamese, where translation artifacts and schema drift further amplify syntactic mismatch.

These limitations collectively motivate the development of ViR², which combines semantic retrieval, POS-based syntactic alignment, and diversity-aware beam-guided re-ranking to provide a more comprehensive solution for low-resource NL2SQL prompting.

2.4. Low-resource NL2SQL challenges

Research on Vietnamese NL2SQL remains fundamentally constrained by the lack of native, large-scale datasets. Current efforts primarily rely on ViText2SQL, a machine-translated version of Spider, which, despite its utility, introduces substantial translation noise such as schema drift, unnatural phrase structures, and mismatches between Vietnamese expressions and database column names [1, 4]. A quantitative breakdown of these translation artefacts is provided in APPENDIX. These issues directly impair the alignment between natural-language queries and SQL semantics, making the Text-to-SQL task more fragile than in English.

Beyond ViText2SQL, several Vietnamese structured-query and structured-reasoning datasets have emerged, but none are native NL2SQL resources. Open-ViTabQA [17] contains 9,911 question-answer pairs over 329 Wikipedia tables spanning 41 domains, and is designed for table-based QA rather than SQL generation. Its questions require complex reasoning over heterogeneous

table structures, including merged cells and multi-row/column alignment, and are evaluated with both answer-level F1 and the Binary Inference Fidelity (BIF) metric to capture logical consistency. Although Open-ViTabQA does not provide SQL annotations, its table-grounded reasoning closely parallels schema-grounded Text-to-SQL and highlights the need for demonstrations that are structurally diverse and robust to noisy tabular layouts—precisely the type of few-shot examples that ViR² aims to select.

Similarly, ViWiQA [18] focuses on Vietnamese open-domain QA over Wikipedia, combining hyperlink-graph-based retrieval with strong reader models on benchmarks such as UIT-ViQuAD and VIMQA. While not SQL-based, ViWiQA demonstrates that Vietnamese systems still struggle with retrieval-augmented reasoning when operating over large, noisy knowledge bases, due to sparse high-quality corpora and limited supervision. This again underscores the importance of robust example selection rather than relying solely on naive semantic similarity, especially in multilingual or translation-imperfect settings.

VIMQA [19] further stresses the complexity of Vietnamese reasoning tasks: it comprises more than 10k question-answer pairs that require multi-hop inference; the reasoning pattern resembles the compositional nature of SQL queries. Moreover, the structural variability in these datasets suggests that ViR²'s lightweight POS-based syntactic modelling and diversity-aware demonstration selection could also be beneficial for table-grounded or multi-hop QA tasks, where structural alignment and reasoning complexity strongly influence model performance through the aggregation of multiple evidence paragraphs, together with sentence-level supporting facts. Although VIMQA operates in an extractive QA paradigm, its multi-step inference model performance. A concise comparison of these Vietnamese datasets is provided in Table 1, highlighting their scale, task formulation, and structural characteristics in relation to NL2SQL.

Table 1. Overview of Vietnamese structured question answering and Text-to-SQL datasets relevant to low-resource natural language to SQL research.

Dataset	Task/Output	Notes
ViText2SQL	Text-to-SQL (SQL queries)	~7k Q-SQL pairs; 166 DBs; machine-translated from Spider; contains translation noise and schema drift.
Open-ViTabQA	Table QA	9,911 QA pairs over 329 Wikipedia tables; 41 domains; includes merged cells, multi-row alignment.
ViWiQA	Open-domain QA	Uses Vietnamese Wikipedia; emphasises retrieval-augmented QA; evaluated on UIT-ViQuAD and VIMQA.
VIMQA	Multi-hop QA	~10k multi-hop questions; requires supporting facts and multi-step inference over multiple paragraphs.

SQL: Structured Query Language; QA: question answering.

Taken together, low-resource Vietnamese NLP faces three compounding issues: (i) scarcity of native annotated NL2SQL datasets, forcing reliance on machine-translated corpora with inherent inconsistencies; (ii) weak cross-domain generalisation, due to limited training resources and translation artifacts that disrupt schema alignment; and (iii) structural and reasoning complexity, evidenced by Vietnamese table QA and multi-hop QA benchmarks, which expose the need for syntactic cues and example-set diversity. These observations align with the design philosophy of ViR², reinforcing its emphasis on semantic-syntactic complementarity and diversity-aware selection as key ingredients for robust question-to-query modelling in low-resource environments. In the following section, we introduce ViR², which operationalises these insights through an unsupervised two-stage framework that integrates semantic retrieval with lightweight syntactic cues and diversity-aware example set construction.

2.5. Search- and ranking-based selection in prompting

Different approaches, including search- and ranking-based methods, have been investigated to improve in-context example selection by balancing semantic relevance and diversity. Techniques such as MMR have been used to reduce redundancy while keeping relevance, and ranking-aware selection has proven effective in retrieval-augmented generation and summarisation [20]. Recent work including PromptRefine [21] shows that embedding-based retrieval along with diversity-aware filtering can select few-shot informative demonstrations effectively, in particular for low-resource languages where high-quality examples are few. PromptRefine utilises example banks of related languages and employs similarity-guided ranking for the refinement of example sets. It presents how cross-lingual retrieval and selection can significantly impact few-shot performance. Meanwhile, RankCoT [22] incorporates ranking into chain-of-thought prompting for the sake of improving multi-step reasoning. Nonetheless, these are effective, without an existing effort in NL2SQL-mainly for low-resource languages-using beam-guided re-ranking which jointly considers semantic relevance, syntactic similarity and intra-set diversity. This gap has directly motivated us to design ViR², which will leverage semantic retrieval followed by beam-guided syntactic-diversity re-ranking, filling the mentioned limit and setting up the stage for the method that is discussed in the next section.

3. ViR²: Two-stage unsupervised example selection

Motivated by the limitations discussed above, we propose ViR² for in-context example selection in NL2SQL, an unsupervised and language-agnostic framework. Unlike existing methods that rely on semantic similarity or heuristic clustering, ViR² combines semantic retrieval with beam-guided re-ranking to jointly optimise semantic relevance, syntactic alignment, and intra-set diversity. Our

method consists of two stages. First, semantic retrieval selects the top-M candidate examples most relevant to the input question. Second, beam-guided search explores and scores multiple example subsets to select the top-K demonstrations for prompting a large language model. This two-stage design enables ViR² to produce structurally diverse and contextually aligned prompts without requiring any labelled data or schema annotations. It is particularly suitable for low-resource and cross-lingual NL2SQL scenarios.

3.1. Problem definition

In few-shot NL2SQL prompting, the model receives a natural language question q and a pool of M candidate examples:

$$C = \{C_1, C_2, \dots, C_M\} \quad (1)$$

where each candidate C_i consists of a question-SQL pair. The objective is to select a subset of K examples:

$$\varepsilon^* = \{E_1, E_2, \dots, E_K\} \subseteq C \quad (2)$$

that best guides a LLM to generate the correct SQL query for q . We formalise example selection as an unsupervised optimisation problem:

$$\varepsilon^* = \arg\max S(\varepsilon, q); \varepsilon \subseteq C; |\varepsilon| = K \quad (3)$$

where $S(\varepsilon, q)$ is a scoring function that evaluates each candidate set based on three complementary criteria:

(1) Semantic relevance: The selected examples should be topically related to the input question.

(2) Syntactic alignment: The examples should exhibit surface-level structural similarity, measured through token-level POS-tag overlap between the question and candidate examples. This lightweight syntactic signal effectively complements semantic cues, consistent with cognitive evidence on the interplay between parsing and meaning construction [23].

(3) Intra-set diversity: The example set should cover a variety of SQL operations to avoid redundant prompts.

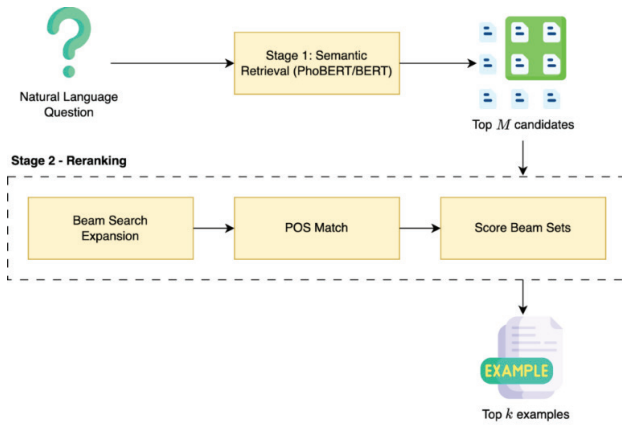


Fig. 1. Vietnamese Intelligent Retrieval and Re-ranking two-stage example selection framework.

This problem formulation is illustrated in Fig. 1, which presents the overall ViR² pipeline, and Fig. 2, which details the beam-guided re-ranking process used to explore and score candidate example sets. By combining semantic retrieval and beam-based search with syntactic and diversity scoring, ViR² efficiently identifies the most informative in-context examples for NL2SQL prompting without requiring any labelled data, schema annotations, or execution feedback.

3.2. Stage 1 - Semantic retrieval

The first stage of ViR² focuses on efficiently narrowing the candidate pool by retrieving the most semantically relevant examples for a given Vietnamese question. Each question and candidate example is encoded using PhoBERT [24], which has demonstrated strong performance in capturing both semantic and syntactic features for Vietnamese. Semantic similarity between the input question q and each candidate C_i is computed via cosine similarity:

$$\text{sim}(q, C_i) = \frac{h_q \cdot h_i}{\|h_q\| \cdot \|h_i\|} \quad (4)$$

where h_q and h_i are the PhoBERT embeddings of the input question and the candidate example.

The top-M examples with the highest similarity scores are retained as the retrieval pool:

$$C_{\text{top-M}} = \text{TopM}(C, \text{sim}(q, C_i)) \quad (5)$$

which is then passed to Stage 2 - Beam-guided re-ranking for structural and diversity-aware selection (Fig. 1).

By design, this semantic retrieval step is lightweight and unsupervised, requiring no schema annotations or SQL execution feedback. While ViR² is developed primarily for Vietnamese NL2SQL, its adaptability to English is demonstrated later in Section 5 through ablation studies.

3.3. Stage 2 - Beam-guided re-ranking

After Stage 1 produces a semantically relevant retrieval pool $C_{\text{top-M}}$, Stage 2 performs beam-guided re-ranking to select the final top-K examples. Instead of relying on greedy selection, beam search systematically explores multiple candidate subsets to balance semantic relevance with structural compatibility. In our framework, this structural signal is operationalised through lightweight token-level POS-tag overlap, which provides robust syntactic cues during example set construction. Such joint semantic-syntactic guidance is essential, as LLMs often fail to make consistent use of long prompts when relevant demonstrations are buried or structurally mismatched [24].

At each step, beam search expands partial sets by adding one new candidate example and evaluates each beam using a composite scoring function:

$$S(\mathcal{E}, q) = \lambda_{\text{sem}} \cdot \text{Sem}(\mathcal{E}, q) + \lambda_{\text{syn}} \cdot \text{Syn}(\mathcal{E}, q) + \lambda_{\text{div}} \cdot \text{Div}(\mathcal{E}) \quad (6)$$

where $\text{Sem}(\mathcal{E}, q)$ means cosine similarity between q and the examples in $\mathcal{E}$; $\text{Syn}(\mathcal{E}, q)$: means syntactic alignment measured via POS-tag overlap. $\text{Div}(\mathcal{E})$ means intra-set diversity, measured by 1- average pairwise cosine similarity among selected examples; $\lambda_{\text{sem}}, \lambda_{\text{syn}}, \lambda_{\text{div}}$ are non-negative weights balancing the three components.

In practice, we fix $\lambda_{\text{sem}} = 1$ and $\lambda_{\text{syn}} = 1$, and tune λ_{div} using a small grid search over $\{0.1, 0.3, 0.5\}$ on a held-out portion of the ViText2SQL development set, selecting $\lambda_{\text{div}} = 0.3$ as the best trade-off between syntactic alignment and diversity. This value is then kept fixed for all experiments in Sections 4-5.

For a question q and a candidate example C_p , we compute:

$$\text{POS_matching}(q, C_i) = \frac{1}{|q|} \sum_{t \in q} 1[\text{POS}(t) \in \text{POS}(C_i)] \quad (7)$$

where $\text{POS}(t)$ is the POS tag of token t and $\text{POS}(C_i)$ denotes the multiset of POS tags appearing in C_i . This normalised overlap score provides a lightweight approximation of syntactic similarity that is robust for short NL2SQL queries. Higher values indicate stronger structural correspondence.

Beam search proceeds layer by layer (see Fig. 2):

- Initialisation: Each beam starts with a single example from C_{top-M} .
- Expansion: Beams are expanded by adding the next candidate example.
- Scoring & Pruning: Each beam ε is scored using $S(\varepsilon, q)$; only the *top-B* beams are kept.
- Termination: After K steps, the best-scored beam ε^* is returned as the final top-K example set.

Formally, Stage 2 optimises the same objective defined in Eq. (3), but the search is constrained to the semantically retrieved pool C_{top-M} . Beam search incrementally expands partial sets and prunes low-scoring beams based on the composite score $S(\varepsilon, q)$, efficiently yielding the final *top-K* examples.

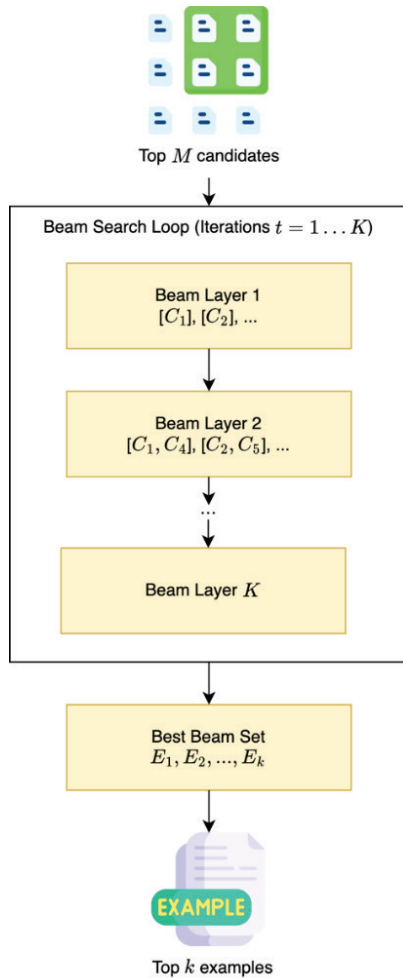


Fig. 2. Beam-guided expand-score-prune loop for final example-set selection in Vietnamese Intelligent Retrieval and Re-ranking.

3.4. Implementation notes

ViR² is implemented as a modular and lightweight example selection framework, designed to integrate seamlessly with NL2SQL pipelines such as ViPERSQL. The framework exposes a unified interface:

```
selected_examples = select(question=q, K=K)
```

which encapsulates both Stage 1: Semantic retrieval and Stage 2: Beam-guided re-ranking. This design allows the system to remain agnostic to the underlying LLM and easily compatible with different prompting strategies, including standard few-shot and CoT prompting.

From an engineering perspective, ViR² has several notable characteristics:

Unsupervised and schema-free: The method does not require any labelled training data, database schema annotations, or SQL execution feedback. This makes it particularly suitable for low-resource settings, where annotated NL2SQL datasets are scarce.

Modular architecture: Embedding models and syntactic parsers are implemented as pluggable components. In our default Vietnamese configuration, we use PhoBERT for semantic encoding and a POS-tagger to obtain the token-level POS tags required for syntactic overlap scoring in Stage 2. These modules can be seamlessly replaced with English or multilingual counterparts, enabling cross-lingual and ablation studies without modifying the core ViR² algorithm (see Section 5).

Computational efficiency: By first narrowing candidates via semantic retrieval and then applying beam-guided pruning, the method avoids the combinatorial explosion of $\binom{M}{K}$ subsets while still achieving high structural coverage.

Extensibility: The framework can integrate additional scoring components (e.g., schema-linking confidence) or alternative search heuristics without altering its core interface.

Combined with the two-stage pipeline illustrated in Fig. 1 and the beam search mechanism shown in Fig. 2, ViR² provides a practical and scalable solution for unsupervised example selection in few-shot NL2SQL prompting. Its design enables straightforward adaptation to new languages

and domains, which is further validated in our experiments and ablation studies in Section 5.

Open-source availability: The full implementation of ViR² and all experiment scripts are publicly released on GitHub*.

4. Experimental setup

We investigated the effectiveness of ViR² for low-resource NL2SQL prompting through experiments conducted in the Vietnamese setting, where labelled datasets are limited, and translation artefacts affect SQL generation. We first evaluate ViR² on the ViText2SQL benchmark in its original target language. Then, we extend experiments to English datasets, including BIRD, to demonstrate cross-lingual generalisation. In this section, the overall experimental setup is summarised, covering datasets, models, prompting strategies, and evaluation metrics.

4.1. Datasets

Research in Vietnamese NL2SQL suffers from the severe lack of data, where no large-scale native datasets like Spider or BIRD exist in English. We present results focusing on ViText2SQL, supplemented with BIRD for some cross-lingual experiments.

ViText2SQL is a Vietnamese Text-to-SQL dataset translated from Spider, which involves around 7k question-SQL pairs across 166 databases. It remains the only publicly available Vietnamese NL2SQL benchmark of reasonable quality. The dataset retains Spider’s cross-domain query diversity while introducing an additional layer of translation noise: schema mismatches and lexical artifacts. In this sense, it forms an ideal benchmark for testing the performance of ViR² under low-resource conditions. However, since it relies on machine translation, ViText2SQL does not actually represent the native linguistic patterns in Vietnamese. We will later discuss it in more detail in our cross-lingual and translation-robustness analysis, Section 5.4.

We adopt the BIRD (dev) split for cross-lingual validation. Since BIRD does not offer a public test set, the dev split is a standard choice for evaluation. In order to avoid overfitting to high-resource

English benchmarks and also maintain consistency with our low-resource setting, we evaluate ViR² on a reproducible subset of 500 queries, sampled using a fixed random seed (seed=42) and uniform sampling per database in order to maintain schema diversity. This controlled sampling strategy ensures that our cross-lingual evaluation reflects structural diversity rather than dataset scale, thereby enabling clearer attribution of performance differences to example-selection behaviour instead of resource imbalance.

To put the Vietnamese resource landscape into perspective, we also draw on three structured-reasoning datasets—Open-ViTabQA, ViWiQA, and VIMQA—though these are not directly used for evaluation. Open-ViTabQA points out, for instance, the challenges of table-grounded reasoning with, among others, merged cells and multi-row alignment. ViWiQA points to retrieval-augmented limitations for Vietnamese open-domain QA, while VIMQA illustrates the difficulty of multi-hop compositional reasoning. Taken together, these limitations highlight the lack of native NL2SQL resources and motivate our experiments to focus on the closest available benchmark, ViText2SQL.

A summary of both datasets is provided in Table 2, and the SQL query type distribution of ViText2SQL is analysed in Table 3 to highlight the skew toward complex queries.

Table 2. Summary of datasets.

Dataset	Language	Origin	#Questions	#Databases
ViText2SQL	Vietnamese	Spider (MT)	7,000	166
BIRD (dev)	English	Native	10,000+	600+

Table 3. Structured Query Language query type distribution in ViText2SQL.

SQL type	Description	Count	Percentage (%)
SET_OP	UNION/INTERSECT/EXCEPT	419	6.1
SUBQUERY	Contains subqueries	764	11.2
JOIN	Multi-table join operations	2,582	37.8
GROUP	Includes GROUP BY	666	9.7
AGG	Aggregation (COUNT, SUM, AVG, etc.)	774	11.3
ORDER	Includes ORDER BY	527	7.7
SELECT_WHERE	Simple SELECT with WHERE	860	12.6
SELECT_ONLY	Simple SELECT without conditions	239	3.5

SQL: Structured Query Language.

*<https://github.com/hoadm-net/ViPERSQL>

The ViText2SQL dataset is heavily skewed toward JOIN queries, with relatively few simple SELECT statements. This pattern underlines the need for diverse example selection to cover a range of SQL structures, as detailed in Table 3.

Overall, ViText2SQL is small, translation-based, and dominated by complex queries, which makes it ideal for evaluating unsupervised example selection in low-resource scenarios. In contrast, BIRD (dev) provides a native English benchmark that we sample partially to demonstrate the cross-lingual compatibility of ViR² with an eye toward full high-resource optimisation.

4.2. Language models and tools

Following the focus of ViR² on Vietnamese NL2SQL, our default configuration uses PhoBERT-base-v2 for semantic embeddings and Underthesea for POS tagging. PhoBERT has been shown to effectively capture Vietnamese lexical and syntactic properties. Underthesea tokenises and brings in POS annotations that are consistent between colloquial and formal Vietnamese. Importantly, neither module requires additional fine-tuning since ViR² operates in a purely unsupervised setting.

To investigate cross-lingual generalisation, we adapt ViR² to English by replacing PhoBERT with BERT-base-uncased for semantic encoding and Underthesea with the lightweight spaCy en_core_web_sm model for POS tagging. We deliberately use the small spaCy model for two reasons: (i) it reduces latency by an order of magnitude during beam-guided re-ranking, and (ii) pilot experiments with medium and large variants show negligible performance differences (<0.3 EM) for our syntactic-overlap scorer, which relies only on coarse token-level POS categories rather than fine-grained dependency structures. This observation aligns with prior findings that in-context retrieval benefits more from high-quality embeddings than from heavy syntactic models.

Table 4 summarises the embedding and parsing tools used in our Vietnamese and English setups.

Table 4. Embedding models and Part of Speech parsers.

Language	Embedding model	POS parser	Usage scope
Vietnamese	PhoBERT-base-v2 (VinAI)	Underthesea	Default ViR ² configuration
English	BERT-base-uncased	spaCy (en_core_web_sm)	Cross-lingual ablation

POS: Part of Speech; ViR²: Vietnamese Intelligent Retrieval and Re-ranking.

For Stage 1 (semantic retrieval), sentence-level embeddings are extracted using the corresponding BERT/PhoBERT model, and cosine similarity is applied to rank candidate examples. For Stage 2, POS tags from the associated parser contribute to syntactic-alignment scoring in the composite function $S(E, q)$ (Eq. 6). This modular architecture allows ViR² to switch between languages without retraining and without any schema-dependent supervision, making it suitable for low-resource NL2SQL scenarios and cross-lingual evaluation.

For SQL generation, we use a single LLM configuration across all experiments. Unless otherwise stated, ViR² generates SQL with the GPT-4.1-mini model (OpenAI, May-2025 snapshot), accessed via the ViPERSQL prompting interface with temperature = 0.1, top-p = 1.0, and a maximum output length of 2,000 tokens. Chain-of-thought decoding is disabled unless explicitly evaluated (e.g., zero-shot CoT in Section 5). Although ViPERSQL supports alternative backends such as GPT-4.1-Turbo and Claude 3.5-Sonnet, these models are not used in the results reported in this article.

4.3. Prompting strategies

We evaluate ViR² under three prompting paradigms for Text-to-SQL: zero-shot, few-shot, and CoT. These settings are chosen to assess both baseline performance and the impact of example selection on low-resource NL2SQL. All few-shot experiments use the same candidate pool size $M=50$, top- $K=3$ examples for prompting, and a beam size of 5 in Stage 2 of ViR², unless otherwise stated.

Zero-shot prompting: The LLM generates SQL queries without any in-context examples. This serves as a lower-bound baseline, generally effective only for frequent or simple query types.

Few-shot prompting: The LLM is provided with K question-SQL example pairs as in-context demonstrations. We evaluate four example selection strategies under identical M and K configurations:

- Random: Uniformly sampled examples.
- Skill-KNN [11]: Retrieval based on skill-oriented descriptions.
- Dicl [12]: Clustering-based selection for structural diversity.
- ViR² (ours): Two-stage selection combining semantic retrieval and beam-guided syntactic-diversity re-ranking.

By using identical M and K across methods, we ensure fair comparison and isolate the impact of example selection.

CoT prompting: We apply CoT prompting to encourage step-by-step reasoning in Text-to-SQL generation. CoT is tested in both zero-shot and few-shot settings, allowing us to evaluate whether structurally diverse examples selected by ViR² further enhance reasoning performance.

To ensure reproducibility, we summarise all LLM-related hyperparameters used in our experiments including model backend, decoding configuration, and maximum output length in Table 5 below. These parameters remain fixed across all prompting strategies and languages unless otherwise specified.

Table 5. Large language model hyperparameters for reproducibility.

Parameter	Value	Notes
LLM backend	GPT-4.1-mini	Used for all SQL generation experiments
Temperature	0.1	Stable decoding for deterministic SQL structure
Top-p	1.0	Default greedy-sampling hybrid
Max output length	2,000 tokens	Ensures full SQL generation without truncation
CoT mode	Disabled unless specified	Enabled only in CoT experiments
Random seed	42	Controls reproducibility of few-shot sampling

LLM: large language model; SQL: Structured Query Language; CoT: Chain-of-Thought.

4.4. Evaluation metrics

Our evaluation is based on the metrics adopted in the original ViText2SQL benchmark, focusing on SQL generation quality rather than execution accuracy due to the lack of public databases. We report three complementary metrics, including EM, Valid SQL Ratio, and Component-level F1, summarised in Table 6. In this regard, EM measures the percentage of the predicted SQL queries exactly matching the reference SQL string after canonical formatting. This metric is strict and sensitive to table alias variations, ordering of conditions, and formatting choices. Valid SQL Ratio measures the proportion of syntactically valid SQL queries that can be parsed by a SQL parser. Reflecting structural correctness, it fails to capture semantic faithfulness.

Table 6. Evaluation metrics.

Metric	Description	Purpose
EM	% of predicted SQL queries identical to gold SQL	Measures full query generation accuracy
Valid SQL ratio	% of syntactically valid and parsable SQL queries	Evaluates structural correctness
F1-score	F1 for major clauses (SELECT, WHERE, GROUP BY, etc.)	Captures partial correctness and coverage

EM: Exact Match; SQL: Structured Query Language.

For instance, Component-level F1 - which has now been better defined - is a more fine-grained assessment by calculating precision and recall over major SQL clauses. According to the ViText2SQL specification, every SQL query is decomposed into clause-level components, such as SELECT, WHERE, GROUP BY, ORDER BY, aggregation functions, JOIN conditions, etc., and F1 is computed against predicted components to gold components. This captures partial correctness even when the full SQL string does not exactly match and thus is more robust to aliasing and formatting differences in low-resource settings.

In addition, because ViR² introduces a two-stage retrieval and re-ranking pipeline, we also measure end-to-end latency (semantic retrieval + re-ranking + LLM inference) to reflect the practical feasibility of each example-selection strategy. Latency is averaged over all queries and reported in seconds.

Together, these metrics jointly evaluate accuracy, structure, and practical usability, providing a fair and reproducible comparison between ViR² and all baselines.

5. Results and analysis

We now report the experimental results of ViR² on both Vietnamese and English NL2SQL benchmarks. This section evaluates the framework across the prompting strategies and example selection methods introduced in Section 4, focusing on low-resource settings and cross-lingual generalisation. First, we report the overall accuracy and validity of SQL generation on ViText2SQL and then compare the baselines on BIRD to show the language-agnostic adaptability. Additional analyses, including ablation studies and latency measurements, provide further insight into the contribution of semantic retrieval, syntactic alignment, and diversity-aware re-ranking.

5.1. Prompting strategy comparison (ViText2SQL)

We start by running several baseline prompting paradigms on ViText2SQL: zero-shot, random few-shot, and CoT prompting. In the few-shot setting, examples are randomly sampled, while CoT is run both without in-context examples (zero-shot CoT) and with random few-shot demonstrations. For completeness, we also include a fine-tuned baseline from the original study about ViText2SQL [4]. The results in terms of EM, Component-Level F1, and latency are in Table 7.

Table 7. Prompting strategy performance on ViText2SQL.

Prompting strategy	EM (%)	Component-level F1 (%)	Latency (s)
Baseline	53.3	74.36	n/a
Zero-shot	10.12	81.50	2.425
Few-shot (Random)	18.33	86.04	3.312
CoT (w/o examples)	6.67	81.63	11.562
CoT (with random examples)	18.21	83.74	11.512

EM: Exact Match; CoT: Chain-of-Thought.

Compared with the fine-tuned baseline, GPT-based prompting exhibits a different performance profile, achieving high component-level F1 but relatively low EM. This gap arises because LLMs often generate semantically correct queries that differ from the reference SQL in table or column aliasing, which reduces exact string matching accuracy. While the zero-shot prompting only achieves 10.12% EM, even random few-shot examples almost double the EM and elevate F1 to 86.04%, which indicates the minimal context helps the model align better with schema structure.

CoT reasoning gives a mixed result. Zero-Shot CoT reaches similar F1 to the standard zero-shot but drops in EM, underlining that reasoning itself cannot solve the schema grounding problem. Furthermore, this setting is not applicable in real life due to its inference latency of 11.562 s/query. When CoT is combined with randomly sampled demonstrations (“CoT with random examples”), EM substantially improves (from 6.67 to 18.21%), but F1 only increases modestly. Again, it confirms that CoT benefits from contextual cues, yet random demonstrations still lack syntactic alignment for robust schema grounding. These observations again point out that semantically rich queries, which are syntactically unaligned, remain a bottleneck; hence, structurally diverse and syntactically compatible few-shot examples are needed to help improve both EM and F1 results when the resource is low.

5.2. Example selection performance

We next evaluate the effect of different example selection strategies for few-shot prompting on ViText2SQL, using a candidate pool size of M=50 and K=3 in-context examples. Four representative baselines are considered-Random, Skill-KNN, ASTRES, and DCL-alongside ViR² (ours), which combines semantic retrieval with beam-guided syntactic-diversity re-ranking.

Figure 3 shows the EM results for all strategies. EM is the most informative metric for this comparison because it directly reflects the model’s ability to generate canonical SQL queries consistent with reference outputs.

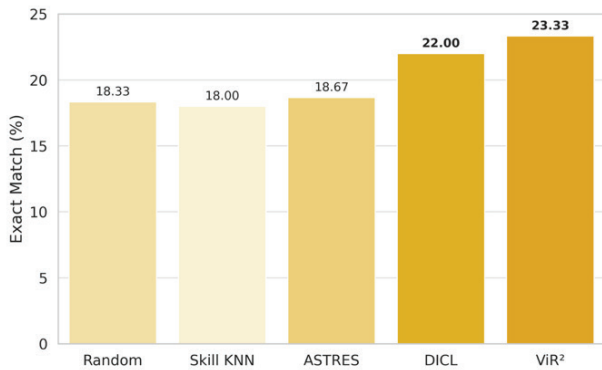


Fig. 3. Exact Match (%) of example selection strategies on ViText2SQL.

From this figure, several observations emerge: First, the highest EM is 23.33% by ViR², which clearly outperforms all baselines, confirming that semantic relevance, syntactic alignment, and intra-set diversity should be combined for effective few-shot prompting in low-resource NL2SQL. DiCL has a rank of second with 22.00%, benefiting from structural clustering but still without global beam-guided optimisation. ASTRES improves over random and Skill-KNN slightly due to its AST-level alignment but lacks diversity-aware re-ranking, thus allowing it only a small EM gain.

In contrast, ViR² consistently finds example sets that are semantically relevant yet structurally complementary, thereby yielding higher EM without prohibitive latency. These results demonstrate that the quality and structural diversity of in-context examples are more critical than quantity, which validates the design choices of ViR² for low-resource Text-to-SQL.

5.3. Ablation study

To analyse the contribution of each component in more detail, we perform an ablation study by removing modules one at a time from the full system. In particular, we assess three ablated variants: removing the POS-based syntactic alignment (“No POS”), removing diversity-aware scoring (“No Diversity”), and replacing beam search with greedy selection (“No Beam”). All experiments are run under identical settings on ViText2SQL with M=50, K=3.

Table 8 shows the ablation results where removal of each module decreases the scores of EM and F1.

Table 8. Ablation study on ViR² components (ViText2SQL).

Model	Exact Match	F1
ViR ²	23.33	87.83
ViR ² No POS	21.67	86.70
ViR ² No Diversity	23.33	87.25
ViR ² No Beam	22.67	87.66

The results show that the full ViR² model achieves the highest EM and F1, confirming the benefit of integrating all components:

- Removing POS-level matching leads to a clear drop in both EM and F1, indicating its importance for aligning examples with the target query syntactically.
- Disabling the diversity module maintains EM but slightly reduces F1, showing that structural variety improves clause-level correctness.
- Replacing beam search with greedy top-K selection causes moderate degradation in both metrics, confirming that beam exploration helps avoid suboptimal example sets.

Overall, each module contributes to the final performance, and the full combination provides the most consistent improvements, validating ViR²'s design as a modular yet synergistic framework for low-resource NL2SQL.

5.4. Cross-lingual generalisation and GPT-translated data analysis

To assess the language-agnostic capability of ViR², we perform two complementary cross-lingual evaluations on the BIRD development set: (i) applying ViR² directly to English queries, and (ii) translating BIRD queries into Vietnamese to simulate a low-resource scenario where English NL2SQL datasets are adapted into Vietnamese. As dataset characteristics have already been described in Section 4.1, we focus here solely on how ViR² behaves under cross-lingual and translation-induced variation.

We evaluate three settings:

(1) ViR² (En):

a) English BIRD dev queries processed using BERT-base-uncased for semantic retrieval and spaCy (en_core_web_sm) for POS tagging.

b) This setting isolates ViR²'s cross-lingual generalisation without involving translation.

(2) ViR²-Random (Vi):

a) Vietnamese queries produced by GPT-4.1-mini translation.

b) In this ablation variant, Stage-1 semantic retrieval is preserved, but Stage-2 syntactic re-ranking is disabled: the final K demonstrations are sampled uniformly at random from the retrieved candidate pool C_{top-M} without applying POS-based alignment or diversity scoring.

c) This baseline isolates the contribution of ViR²'s re-ranking module and shows how translated NL2SQL data behaves without structured example selection.

(3) ViR² (Vi):

a) The same GPT-translated Vietnamese queries evaluated with the full ViR² pipeline-PhoBERT embeddings + Underthesea POS tags.

b) This setting tests whether ViR² can mitigate the instability introduced by machine translation.

Translation process: We translate each English query into Vietnamese using GPT-4.1-mini with a constrained format preserving SQL-linked entities whenever possible. Although high-quality, the translated queries frequently exhibit schema drift, unnatural phrasing, or inconsistent terminology relative to column names-issues consistent with those previously observed in ViText2SQL. A detailed quantitative audit of these translation artefacts is provided in APPENDIX. This analysis allows us to specifically attribute performance drops to translation-induced variability rather than dataset-specific factors already covered earlier.

Table 9 reports the EM and Component-F1 results for all three cross-lingual configurations.

Table 9. Cross-lingual and GPT-translated performance on Big Bench for Large-scale Database Grounded Text-to-SQL Evaluation (Exact Match and F1 in %).

Model	Exact Match	F1
ViR ² (En)	1.78	79.25
ViR ² Random (Vi)	4.32	79.43
ViR ² (Vi)	1.26	78.57

ViR²: Vietnamese Intelligent Retrieval and Re-ranking.

These results yield three main observations: First, ViR² generalises to English with 1.78% EM and 79.25% F1 under the setting of ViR² (En). Although low in absolute EM-consistent with the difficulty of BIRD-this confirms that ViR²'s semantic-syntactic scoring framework transfers across languages with minimal modifications. Second, GPT-translated Vietnamese queries perform poorly under random example selection with 4.32% EM despite a comparable component-level F1 to English. This indicates that machine translation introduces structural inconsistencies that disproportionately affect EM. Third, applying ViR² on translated Vietnamese queries results in an EM drop to 1.26%, which suggests that translation artifacts hinder both interpretation and example-selection quality. However, the relatively stable F1 score across all settings indicates that ViR² maintains clause-level correctness even when translation noise disrupts global SQL structure. These findings together highlight that: (i) ViR² is language-agnostic; (ii) translation-based data augmentation is not reliable for NL2SQL; and (iii) high-quality, native Vietnamese resources continue to remain indispensable for robust Text-to-SQL research.

6. Discussion

Our experiments provide important insights into the effectiveness and generalisability of ViR². First, the results on ViText2SQL indicate that ViR² outperforms both the baseline prompting strategies and the existing example selection methods. This convincingly demonstrates that the combination of semantic retrieval, POS-based syntactic alignment, diversity scoring, and beam-guided re-ranking forms an effective approach to few-shot Text-to-SQL in low-resource settings. These ablation studies further indicate that each component contributes to overall performance and that syntactic alignment and beam exploration are the most important factors for improving EM accuracy.

Another exciting insight comes from the cross-lingual experiments: ViR² can be easily adapted to English, for which only PhoBERT and Underthesea are replaced with BERT and spaCy, respectively,

proving the language-agnostic capabilities of the framework. Simultaneously, experiments on GPT-translated BIRD queries show that automatic translation introduces schema mismatch, semantic drift, and unnatural phrasing, significantly lowering the scores of EM despite the relatively stable component-level F1 scores. The results demonstrate that machine translation cannot replace high-quality native NL2SQL datasets in low-resource settings.

Despite these promising findings, several limitations still remain: ViR² relies on pre-trained embeddings and POS parsers that can inherit domain or language biases. Furthermore, the present framework does not make use of schema execution feedback nor database content; as a result, it cannot disambiguate queries or resolve aliasing automatically. Overall, these observations indicate that careful in-context example selection and language-sensitive prompt engineering are crucial for building reliable Text-to-SQL systems in low-resource and multilingual settings.

7. Conclusions and future work

This article introduced ViR², a two-stage unsupervised framework for example selection in few-shot Text-to-SQL, tailored for low-resource and multilingual scenarios. ViR² first performs semantic retrieval to select candidate examples and then applies beam-guided re-ranking that considers both POS-based syntactic alignment and structural diversity. Furthermore, ViR² is integrated into our unified Text-to-SQL framework, ViPERSQL, which provides a modular and extensible platform that supports multiple retrieval strategies, embedding backbones, and prompting configurations, thereby facilitating reproducible research and practical deployment across languages. Extensive experiments in ViText2SQL prove that ViR² consistently achieves higher EM and component-level F1 than baseline prompting and existing example selection methods. Ablation studies confirm the contribution of each module to performance, and cross-lingual evaluations demonstrate that ViR² generalises to English. Furthermore, experiments with GPT-translated Vietnamese queries show

that automatic translations introduce schema and semantic inconsistencies, which again highlights the importance of high-quality contextual examples when resources are limited.

Despite these modest cross-lingual scores, ViR² remains practically valuable because it operates without supervision, schema annotations, or language-specific resources and can therefore function reliably even with noisy or translated data. Its ability to consistently improve example selection under such constraints makes it directly applicable to real-world multilingual NLIDB systems where high-quality native Text-to-SQL data are often absent.

Future work will focus on three directions.

First, we aim to incorporate schema-aware scoring and execution feedback to further improve query disambiguation and handle aliasing automatically.

Second, we plan to explore adaptive beam and diversity control, allowing the framework to dynamically adjust to different database schemas and query complexity.

Finally, lightweight multilingual adaptation and embedding alignment will be investigated to enhance cross-lingual performance without requiring high-quality parallel datasets.

By combining language-agnostic design, structural reasoning, and unsupervised example selection, ViR² provides a practical and extensible approach for robust Text-to-SQL in low-resource and multilingual environments, forming a solid foundation for future research in multilingual natural language interfaces to databases.

APPENDIX: Quantitative Audit of Translation Noise in ViText2SQL

To assess how translation artefacts may affect Text-to-SQL performance, we conducted a lenient-criteria audit of 700 randomly sampled questions ($\approx 10\%$ of the ViText2SQL training split). Only severe and unambiguous issues were counted—minor synonym choices or stylistic differences were not considered errors.

Issue type	Rate	Description
<i>Schema Drift (SCH)</i>	13.9%	Question uses natural Vietnamese terms that do not align with mixed-language schema names.
<i>Entity Mismatch (ENT)</i>	5.3%	EN-VI inconsistency in literal values (titles, room names, product names).
<i>Structural Mismatch (STR)</i>	5.0%	Question phrasing underspecifies or misleads the SQL structure.
<i>Lexical Issues (LEX)</i>	0.4%	Minor typos or awkward phrasing caused by translation.
<i>No Issues (OK)</i>	81.3%	Translation is good and semantically faithful.

These findings show that ViText2SQL remains broadly usable but contains non-trivial translation artefacts (18.7%) that can affect schema alignment and structural interpretation. This supports our argument that native datasets are preferable, and that ViR²'s syntactic-aware example selection helps mitigate such inconsistencies.

Representative examples from the audit

Question: “Cho biết trang trí trong phòng ‘Ẩn dật và thách thức’”

SQL: WHERE tên_phòng = “Recluse and defiance”

Issue: Entity translated inconsistently (VI → EN).

Question: “Tìm những người bạn có giới tính nữ của Alice.”

SQL: ... WHERE t2.tên = “Alice” AND t1.giới_tính = “female”

Issue: “những người bạn” (colloquial) vs schema column “bạn_bè”.

→ Natural phrasing mismatches schema terminology.

These examples demonstrate how translation noise appears in practice: schema vocabulary drift, EN-VI entity inconsistency, and discrepancies between question phrasing and SQL complexity.

Implication for ViR²

Because many errors involve lexical variation but preserved syntactic structure, ViR²'s POS-pattern matching and diversity-aware re-ranking help reduce the impact of these inconsistencies-reinforcing the motivation for our design.

CRedit author statement

Dinh Minh Hoa: Conceptualisation, Methodology, Investigation, Writing original draft; Tran Thien Khai: Supervision, Methodology Guidance, Writing - Reviewing & Editing.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] T. Yu, R. Zhang, K.C. Yang, et al. (2018), “Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and Text-to-SQL task”, *Proceedings of The 2018 Conference on Empirical Methods in Natural Language Processing*, pp.3911-3921, DOI: 10.18653/v1/D18-1425.
- [2] J. Yi, G. Chen, X. Zhou (2025), “Decoupling SQL query hardness parsing for Text-to-SQL”, *Neurocomputing*, **621**, 13pp, DOI: 10.1016/j.neucom.2024.129293.
- [3] T.B. Brown, B. Mann, N. Ryder, et al. (2020), “Language models are few-shot learners”, *Advances in Neural Information Processing Systems*, pp.1877-1901.
- [4] A.T. Nguyen, M.H. Dao, D.Q. Nguyen (2020), “A pilot study of Text-to-SQL semantic parsing for Vietnamese”, *Findings of The Association for Computational Linguistics: EMNLP 2020*, pp.4079-4085, DOI: 10.18653/v1/2020.findings-emnlp.364.
- [5] M. Pourreza, D. Rafiei (2023), “DIN-SQL: Decomposed in-context learning of Text-to-SQL with self-correction”, *Advances in Neural Information Processing Systems*, pp.36339-36348.
- [6] Z. Hong, Z. Yuan, Q. Zhang, et al. (2025), “Next-generation database interfaces: A survey of LLM-based Text-to-SQL”, *IEEE Transactions on Knowledge and Data Engineering*, **37(12)**, pp.7328-7345, DOI: 10.1109/TKDE.2025.3609486.
- [7] D. Zhou, N. Schärli, L. Hou, et al. (2023), “Least-to-most prompting enables complex reasoning in large language models”, *International Conference on Learning Representations 2023*, 61pp.

- [8] Z. Tan, X. Liu, Q. Shu, et al. (2024), "Enhancing Text-to-SQL capabilities of large language models through tailored prompts", *Proceedings of The 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pp.6091-6109.
- [9] B. Zhai, C. Xu, Y. He, et al. (2025), "Optimising reasoning for Text-to-SQL with execution feedback", *Findings of The Association for Computational Linguistics: ACL 2025*, pp.19206-19218, DOI: 10.18653/v1/2025.findings-acl.982.
- [10] Y. Gan, X. Chen, J. Xie, et al. (2021), "Natural SQL: Making SQL easier to infer from natural language specifications", *Findings of The Association for Computational Linguistics: EMNLP 2021*, pp.2030-2042, DOI: 10.18653/v1/2021.findings-emnlp.174.
- [11] S. An, B. Zhou, Z. Lin, et al. (2023), "Skill-based few-shot selection for in-context learning", *Proceedings of The 2023 Conference on Empirical Methods in Natural Language Processing*, pp.13472-13492, DOI: 10.18653/v1/2023.emnlp-main.831.
- [12] J. Kapuriya, M. Kaushik, D. Ganguly, et al. (2025), "Exploring the role of diversity in example selection for in-context learning", *Proceedings of The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.2962-2966, DOI: 10.1145/3726302.3730194.
- [13] Z. Shen, P. Vougiouklis, C. Diao, et al. (2024), "Improving retrieval-augmented Text-to-SQL with AST-based ranking and schema pruning", *Proceedings of The 2024 Conference on Empirical Methods in Natural Language Processing*, pp.7865-7879, DOI: 10.18653/v1/2024.emnlp-main.449.
- [14] R. Luo, L. Wang, B. Lin, et al. (2024), "PTD-SQL: Partitioning and targeted drilling with LLMs in Text-to-SQL", *Proceedings of The 2024 Conference on Empirical Methods in Natural Language Processing*, pp.3767-3799, DOI: 10.18653/v1/2024.emnlp-main.221.
- [15] D. Gao, H. Wang, Y. Li, et al. (2024), "Text-to-SQL empowered by large language models: A benchmark evaluation", *Proceedings of The VLDB Endowment*, **17(5)**, pp.1132-1145, DOI: 10.14778/3641204.3641221.
- [16] C. Mai, R. Tal, T. Mohamed (2024), "Learning metadata-agnostic representations for Text-to-SQL in-context example selection", *NeurIPS 2024 Third Table Representation Learning Workshop*, 23pp.
- [17] D.H. Dao, N.T.K. Huynh, K.Q. Tran, et al. (2025), "Open-ViTabQA: A novel benchmark for Vietnamese question answering on open domain wikipedia table", *Knowledge-Based Systems*, **330(A)**, DOI: 10.1016/j.knosys.2025.114391.
- [18] D.H. Nguyen, N.K. Le, L.M. Nguyen (2023), "ViWiQA: Efficient end-to-end Vietnamese Wikipedia-based open-domain question-answering systems for single-hop and multi-hop questions", *Information Processing & Management*, **60(6)**, DOI: 10.1016/j.ipm.2023.103514.
- [19] N.K. Le, D.H. Nguyen, T. Le, et al. (2023), "VIMQA: A Vietnamese dataset for advanced reasoning and explainable multi-hop question answering", *Proceedings of The Thirteenth Language Resources and Evaluation Conference*, pp.6521-6529.
- [20] X. Wang, Z. Chen, T. Xu, et al. (2024), "In-context former: Lightning-fast compressing context for large language model", *Findings of The Association for Computational Linguistics: EMNLP 2024*, pp.2445-2460, DOI: 10.18653/v1/2024.findings-emnlp.138.
- [21] S.S. Ghosal, S. Pal, K. Mukherjee (2025), "PromptRefine: Enhancing few-shot performance on low-resource indic languages with example selection from related example banks", *Proceedings of The 2025 Conference of The Nations of The Americas Chapter of The Association for Computational Linguistics: Human Language Technologies*, **1**, pp.351-365, DOI: 10.18653/v1/2025.naacl-long.17.
- [22] M. Wu, Z. Liu, Y. Yan, et al. (2025), "RankCoT: Refining knowledge for Retrieval-Augmented Generation through ranking Chain-of-Thoughts", *Proceedings of The 63rd Annual Meeting of The Association for Computational Linguistics*, **1**, pp.12857-12874, DOI: 10.18653/v1/2025.acl-long.629.
- [23] Y. Zhang, M. Taft, J. Tang, et al. (2024), "Neural correlates of semantic-driven syntactic parsing in sentence comprehension", *NeuroImage*, **289**, DOI: 10.1016/j.neuroimage.2024.120543.
- [24] D.Q. Nguyen, A.T. Nguyen (2020), "PhoBERT: Pre-trained language models for Vietnamese", *Findings of The Association for Computational Linguistics: EMNLP 2020*, pp.1037-1042, DOI: 10.18653/v1/2020.findings-emnlp.92.

A wideband and high-gain MIMO microstrip antenna using a frequency selective surface for WLAN applications

Ngoc-Lan Nguyen^{1*}, Hien-Thuan Duong²

¹Posts and Telecommunications Institute of Technology, 11 Nguyen Dinh Chieu Street, Ho Chi Minh City, Vietnam

²Faculty of Engineering and Technology, Sai Gon University, 273 An Duong Vuong Street, Ho Chi Minh City, Vietnam

Received 24 March 2026; revised 25 March 2026; accepted 29 May 2026

Abstract:

This article presents the design of a compact Wireless Local Area Network (WLAN) antenna system tailored for Multiple-Input Multiple-Output (MIMO) applications at 5.3 GHz. The proposed configuration consists of two 2×1 subarrays, in which the integration of a frequency selective surface (FSS) is shown to play a critical role in expanding the antenna's operating frequency range. In particular, this work demonstrates that a simple FSS-based approach can be effectively utilised to achieve significant bandwidth enhancement while maintaining structural simplicity. By incorporating the FSS, the antenna's impedance bandwidth is significantly extended, increasing from 2 to 27% without increasing structural complexity. The design also maintains an isolation level better than 20 dB across the operating frequency band, which is sufficient to suppress mutual coupling between the elements. Furthermore, the design demonstrated a peak measured gain of 9.2 dBi, confirming the suitability of the antenna for high-performance WLAN applications. Moreover, the proposed antenna achieves well-balanced performance in terms of bandwidth, gain, and isolation, making it a practical solution for compact MIMO systems. The design and optimisation of the antenna were carried out using CST Microwave Studio, and its performance was also validated through measurement, thereby confirming the efficacy of the design approach.

Keywords: aperture-coupled feeding, microstrip antenna, MIMO antenna, wireless local area network.

1. Introduction

The rapid development of wireless communication technologies has led to the demand for higher data transmission rates, higher reliability and better spectrum efficiency. Multiple-Input Multiple-Output (MIMO) antenna systems have been proposed as an ideal solution to these problems. Using spatial multiplexing techniques, it can increase the efficiency of the system without extra consumption of spectral resources [1].

MIMO systems provide better data transmission rates but the efficiency of MIMO systems is often degraded due to the mutual coupling between the radiating elements as it can have a significant effect on the overall system performance [2, 3]. Therefore, a requisite factor for the efficient operation of MIMO systems is significant isolation between components.

A range of MIMO antennas with improved isolation performance has been reported in the literature [4-12]. One of the most popular approaches is to etch slots of various geometries on the ground plane to suppress undesired coupling [5, 6]. Furthermore, integrating parasitic elements [11, 12] or other methods [4, 7-10] has also been demonstrated to enhance isolation. Nevertheless, a limited bandwidth of less than 8% remains a major drawback of these proposals.

In recent years, a variety of techniques have been introduced to improve antenna performance, among which the application of metasurfaces has emerged as one of the most common approaches [13, 14]. Based on their functional mechanisms, metasurfaces can be classified into the following categories: frequency-selective surfaces (FSS), high-impedance

*Corresponding author: Email: nnlan@moet.edu.vn

surfaces, perfect absorbers, and reflecting surfaces [15]. Due to their flexible characteristics, which allow for either the suppression or support of wave propagation in a specified frequency range, FSSs have been widely applied, especially in antenna engineering, including: gain enhancement [16–18]; bandwidth extension [19, 20]; and isolation improvement [21, 22]. However, the designs introduced in [18, 21, 22] are restricted by low gain values (below 7 dBi), while other proposals in [17, 19, 20] exhibit a narrow operational bandwidth (less than 11%). Additionally, the large physical size is a limitation of the design in previous study [16].

In this article, a low-profile MIMO antenna with high isolation for WLAN applications at the centre frequency of 5.3 GHz is presented. The proposed design consists of two sets of 2x1 patch elements, which are excited by the aperture-coupled feeding technique etched on the ground surface of the feeding network. In addition, the presence of FSS helps to extend the antenna's operating frequency range. A prototype was fabricated to validate the design concept, and the results indicate that the proposed design achieves an impedance bandwidth of 27% (4.91–6.4 GHz), a peak gain of 9.2 dBi, and isolation above 20 dB (measured). Moreover, the radiation efficiency of the antenna in the operational band is greater than 61%.

2. Design of the proposed MIMO antenna

2.1. An individual antenna element

The design process for the proposed MIMO antenna begins with the development of a single radiating element, the geometry of which is shown in Fig. 1. The antenna has a center frequency of 5.3 GHz for wireless local area network applications. To excite the microstrip antennas, various techniques have been developed such as microstrip line feeds [23], coaxial probes [24], and aperture-coupled configurations [25].

The aperture-coupled feeding technique is chosen as the best one to ensure wideband performance because of its broadband properties. The configuration comprises of a FSS with 3x3 array of unit cells, patch radiator, ground plane (GND), 50 Ω feed line and three dielectric substrates. The feed line and ground are on the opposite sides of Sub. 1, and the patch element is on top of Sub. 2. The FSS layer is

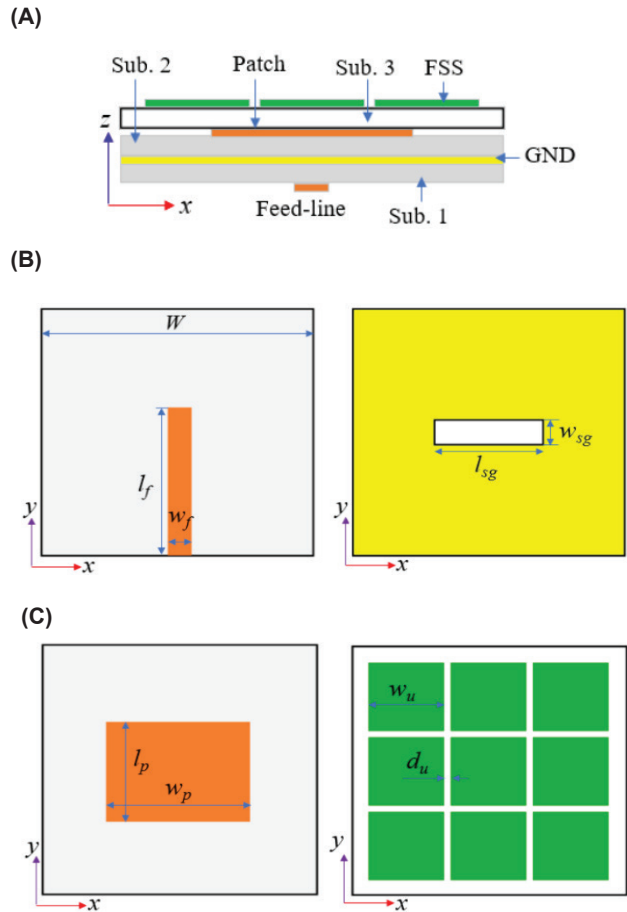


Fig. 1. The configuration of the individual element: (A) cross-section view; **(B)** the feeding line (left) and ground with a slot (right); **(C)** patch (left) and FSS (right). Source: Proposed by authors.

built on sub. 3. Sub. 1 and 2 are made of R04350B material ($\epsilon_r=3.66$, $\tan\delta=0.0037$, $h=1.524$ mm), while FR-4 ($\epsilon_r=4.4$, $\tan\delta=0.02$, $h=1.6$ mm) is utilised as Sub. 3. It should be noted that the FSS is used to extend the operating frequency range. The substrate and patch sizes are 40x40 mm and 22x15 mm, respectively. The antenna was characterised using CST Studio Suite, and its performance is shown in Fig. 2. The optimised design parameters are summarised as follows: $W=40$, $w_f=3.5$, $l_f=23.5$, $l_s=13$, $w_s=3$, $w_p=22$, $l_p=15$, $w_u=10$, $d_u=1$ (unit: mm). The impedance bandwidth, defined at a -10 dB reflection coefficient, spans approximately 20%, covering the range from 5.08 to 6.14 GHz. The antenna demonstrates a peak gain of approximately 7.5 dBi. As depicted in Fig. 3, the radiation characteristics in the x-z and y-z planes confirm a clear broadside radiation pattern. Moreover, the antenna maintains a low side-lobe level below -10 dB.

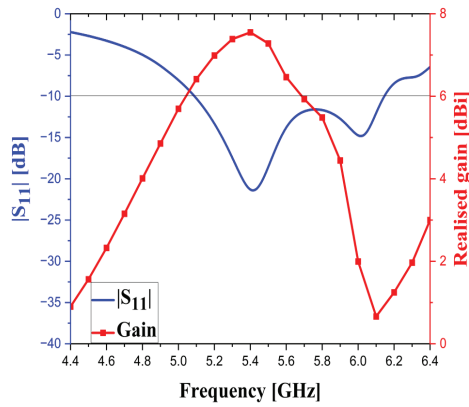
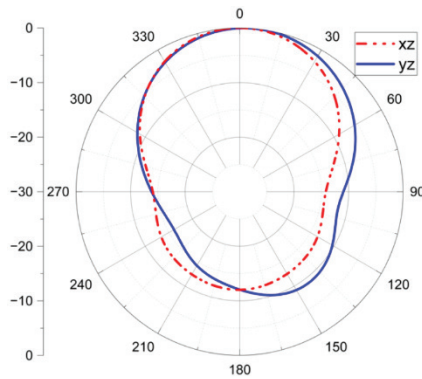


Fig. 2. Reflection coefficient and gain of the individual element. Source: Proposed by authors.

(A)



(B)

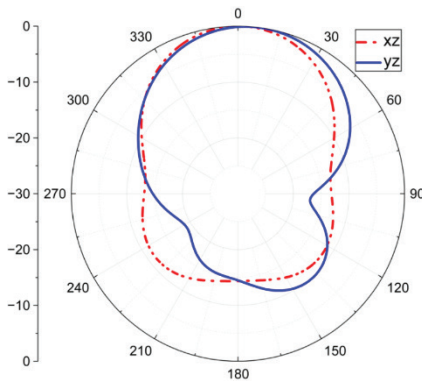


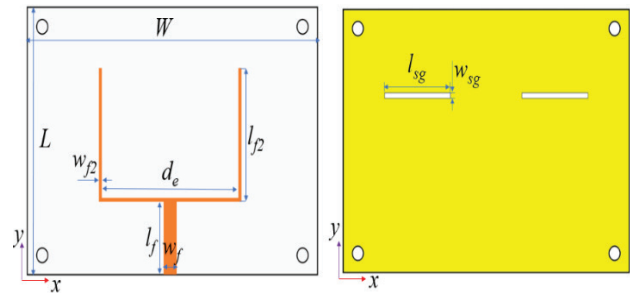
Fig. 3. Radiation pattern of the individual element in the x-z and y-z planes at 5.3 GHz (A) and 5.5 GHz (B). Source: Proposed by authors.

2.2. 2×1 antenna array

Based on the single-element design presented in the previous section, an antenna array has been developed, and its model is shown in Fig. 4. The proposed antenna is printed on a pair of RO4350B substrates, each having a thickness of 1.524 mm

and a relative permittivity of $\epsilon_r=3.66$, with an FR-4 dielectric layer of 1.6 mm thickness utilised for the FSS. Power is distributed to the radiating elements through a T-junction divider. To couple the energy from the divider to the elements, a ground plane with carefully designed slots is inserted between them, enabling efficient transmission across the layers. On the upper side, an FR-4 substrate incorporating a FSS is employed. Figs. 5 and 6 exhibit the reflection coefficient, gain, and radiation pattern of the 2×1 antenna array, respectively. As shown in Fig. 5, the antenna, with impedance matching below -10 dB, achieves a bandwidth of 5.09-6.01 GHz (corresponding to 17.4%) and a gain fluctuation of 5.4-9.4 dBi. Meanwhile, Fig. 6 shows the characteristics of broadside radiation and a side-lobe level (SLL) below -10 dB. In addition, the antenna maintains an efficiency above 65% in the operating frequency band. The final design parameters of the antenna are listed as follows: $W=78$, $L=55$, $w_f=3.5$, $l_f=15$, $d_e=37.5$, $w_{f2}=0.8$, $l_{f2}=27.5$, $w_{sg}=1.2$, $l_{sg}=17$, $w_p=26$, $l_p=16$, $w_u=11.5$, $d_x=3$, $d_y=2.5$ (unit: mm).

(A)



(B)

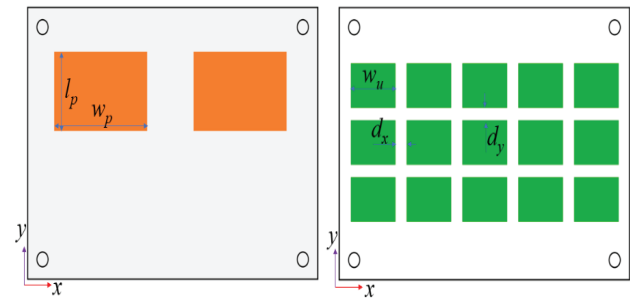


Fig. 4. The structure of the 2×1 antenna array. (A) Power divider (left) and ground layer (right); (B) Patches (left) and FSS (right). Source: Proposed by authors.

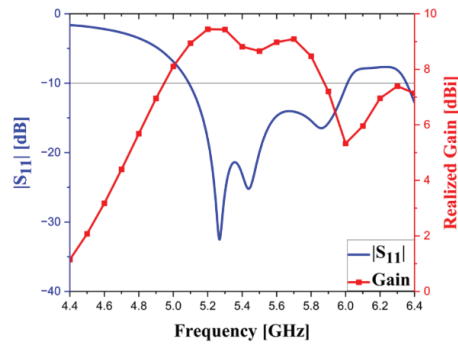


Fig. 5. The reflection coefficient and gain of the 2×1 array. Source: Proposed by authors.

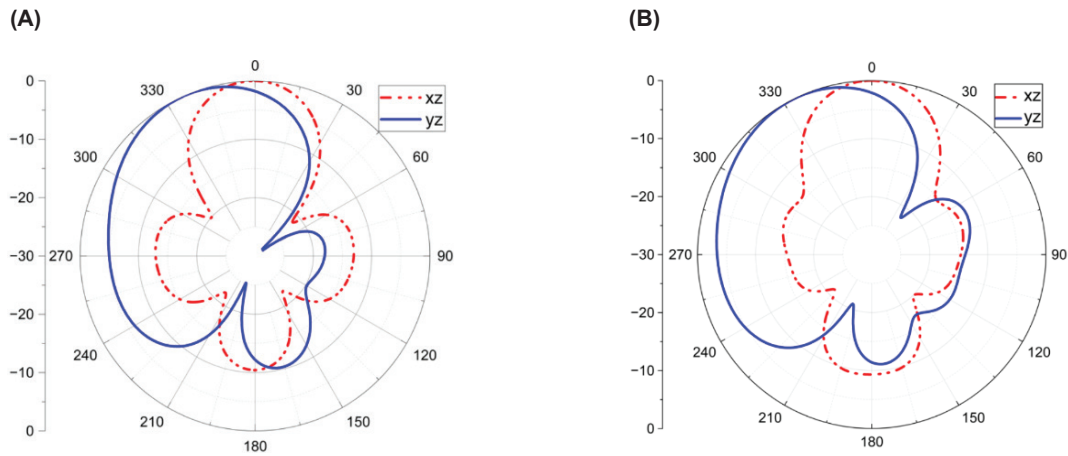


Fig. 6. The radiation characteristics of the 2×1 array in the x-z and y-z planes at 5.3 (A) and 5.5 GHz (B). Source: Proposed by authors.

2.3. MIMO antenna

Following the completion of the 2×1 antenna array design, a MIMO configuration was developed, and the model of the proposed MIMO antenna is

shown in Fig. 7. The proposed structure consists of two identical 2×1 sub-arrays, arranged with an element spacing of 37.5 mm (centre-to-centre). The antenna is implemented on three dielectric layers: Subs. 1 and 2 are Rogers RO4350B ($\epsilon_r=3.66$, thickness=1.524 mm, $\tan\delta=0.0037$), while Sub. 3 is FR-4 ($\epsilon_r=4.4$, thickness=1.6 mm, $\tan\delta=0.02$). The T-junctions and ground planes are printed on either side of Sub. 1, whereas the radiating patches are positioned on the top surface of Sub. 2, and FSS is realised on Sub. 3. The FSS is constructed as a 13×5 grid of unit cells. Following the optimisation, the

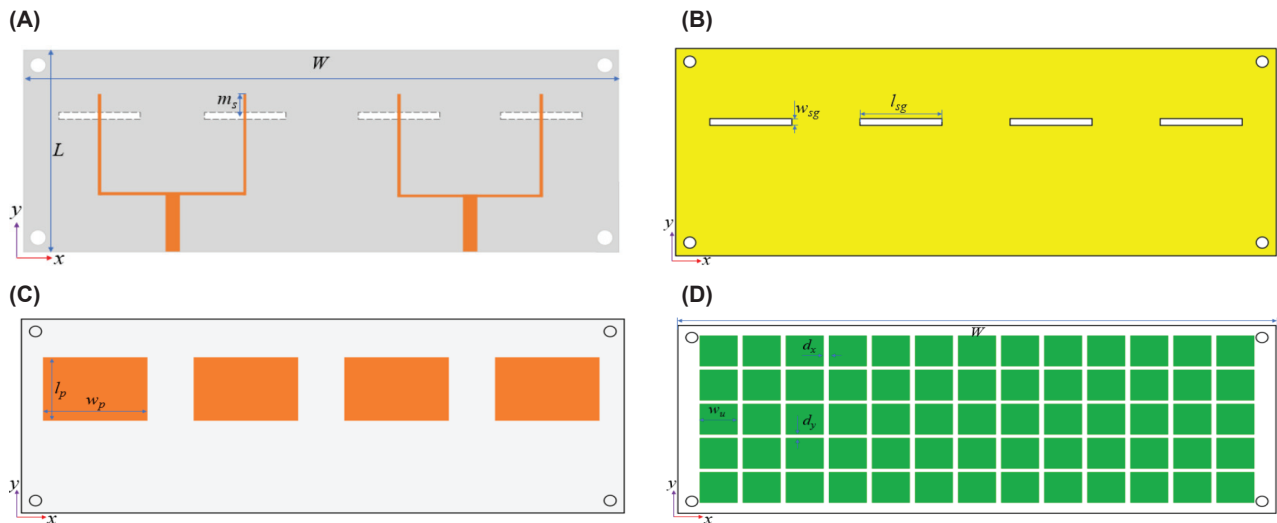


Fig. 7. The model of the proposed MIMO antenna. (A) Feeding; (B) The ground layer; (C) Radiation elements; (D) FSS. Source: Proposed by authors.

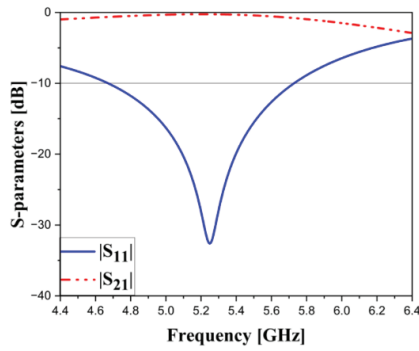


Fig. 8. The characteristics of the FSS. Source: Proposed by authors.

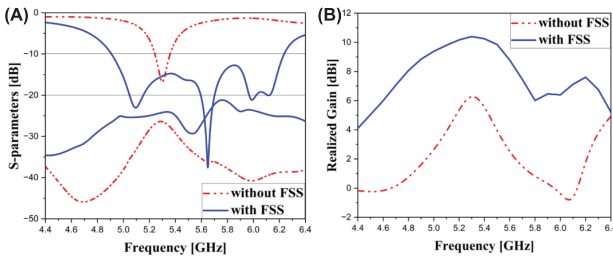


Fig. 9. The effect of the FSS on the S-parameters (A) and the gain (B) of the antenna. Source: Proposed by authors.

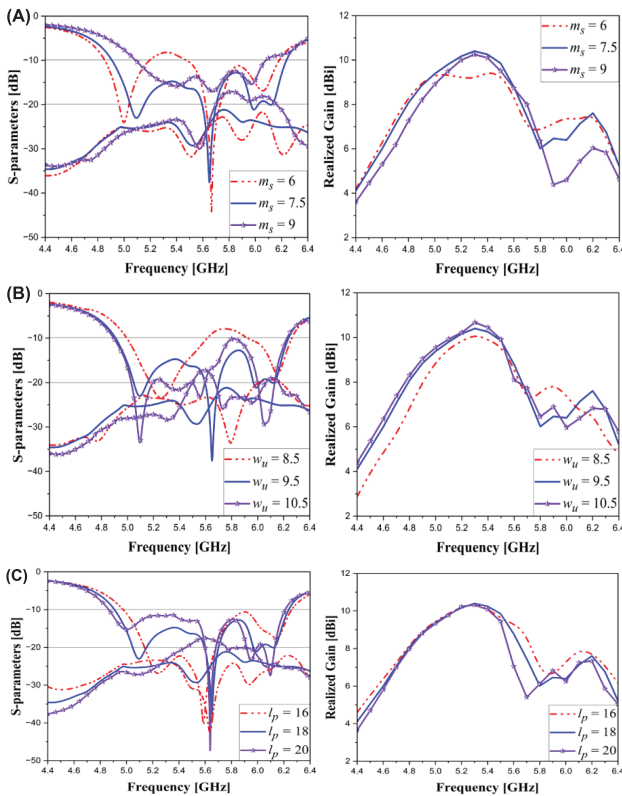


Fig. 10. Parametric analysis of the MIMO antenna. (A) The spacing between the ground slot and the feed line (m_s); **(B)** The unit-cell dimension of the FSS (w_u); **(C)** The patch length (l_p). Source: Proposed by authors.

The integration of the FSS plays a decisive role in extending the operational bandwidth of the antenna. To examine this, a series of full-wave simulations was carried out, with the results presented in Figs. 8 and 9. In Fig. 8, the FSS exhibits a wideband transmission profile, allowing stable operation in the 5 GHz frequency range [26]. A comparison is presented in Fig. 9 to demonstrate the effect of the FSS. As can be seen, the presence of the FSS increases the impedance bandwidth and gain from 2.3 (5.24-5.36 GHz) to 25.3% (4.9-6.24 GHz) and from 6.6 to 10.5 dBi, respectively. The mechanism of performance enhancement through the use of the FSS can be explained as follows: the presence of the FSS introduces additional resonant modes that interact with the inherent resonance of the antenna. This interaction leads to the excitation of multiple resonances across the operating band. Specifically, with the inclusion of the FSS, at least four distinct resonant modes can be identified, whereas only a single resonance is observed in the absence of the FSS. As a result, the impedance bandwidth of the antenna is broadened. Regarding the gain improvement, the improved impedance matching maintained over a wider frequency range leads to an enhancement in the radiation efficiency of the antenna. In addition, the FSS structure contributes to current redistribution in the antenna, leading to better radiation characteristics. As a result, a higher realised gain is achieved compared to the case without the FSS. This not only confirms the bandwidth increase but also shows the improved performance of the antenna.

3. Parametric study

This section mainly discusses the structural factors which decide the performance of the MIMO antenna system. These parameters are critical to achieve stable impedance matching, improve gain and to isolate effectively. The study is concentrated on three important parameters, namely the distance between the ground slot and the feed line (m_s), the unit-cell size of the FSS (w_u) and the patch length (l_p) as shown in Fig. 10. The feeding method is the aperture-coupled one as described earlier. In this technique, energy is coupled to the radiating elements by electromagnetic coupling through slots cut in the ground plane. Thus, the variation in the slot dimensions (m_s) directly affects the impedance

matching features of the antenna. As shown in Fig. 10(A), when $m_s=6$, $|S_{11}|$ does not remain below -10 dB across 5.2-5.45 GHz, indicating weak matching performance. By contrast, at $m_s=9$, although the matching improves, the mutual coupling rises above -20 dB in the 5.7-6.15 GHz band. In terms of gain, the maximum value achieved with $m_s=6$ is approximately 9.5 dBi, which is notably lower than in the other two cases.

Moving to Fig. 10(B), variations in the unit-cell dimensions of the FSS alter its equivalent inductive-capacitive (LC) response. As the unit-cell dimensions decrease, the resonant frequency shifts to higher values, while larger unit cells lead

to a lower frequency shift. It can be observed that for $w_u=8.5$ and 10.5 , $|S_{11}|$ and $|S_{21}|$ exceed -10 dB and -20 dB within 5.61-5.91 GHz and 5.5-5.65 GHz, respectively, while the gain difference in these cases is marginal.

Finally, Fig. 10(C) shows that when $l_p=20$, isolation cannot be maintained above 20 dB in the 5.45-5.75 GHz range. On the other hand, reducing l_p to 16 not only results in a narrower bandwidth but also significantly degrades the impedance matching depth at 5.85-5.95 GHz. In this context, variations l_p not only influence the resonant frequency but also modify the behaviour of surface-wave propagation and near-field interactions, which in turn affect isolation.

4. Simulated and measured results

For experimental validation, the antenna prototype was fabricated and its performance was carefully measured. The realised structures of the antenna array and the MIMO antenna, depicted in Fig. 11, occupy physical dimensions of $78 \times 55 \times 4.8$ mm and $150 \times 55 \times 4.8$ mm, respectively. The S-parameters and gain characteristics of the single antenna (configured as a 2×1 array) and the MIMO antenna are presented in Fig. 12, while the radiation patterns are shown in Figs. 13 and 14. For the single-antenna configuration, both the simulated and measured results indicate an impedance bandwidth (defined at -10 dB) of approximately 17% (5.09-6.01 GHz) and 19.5% (5.25-6.32 GHz), respectively. Within this operating

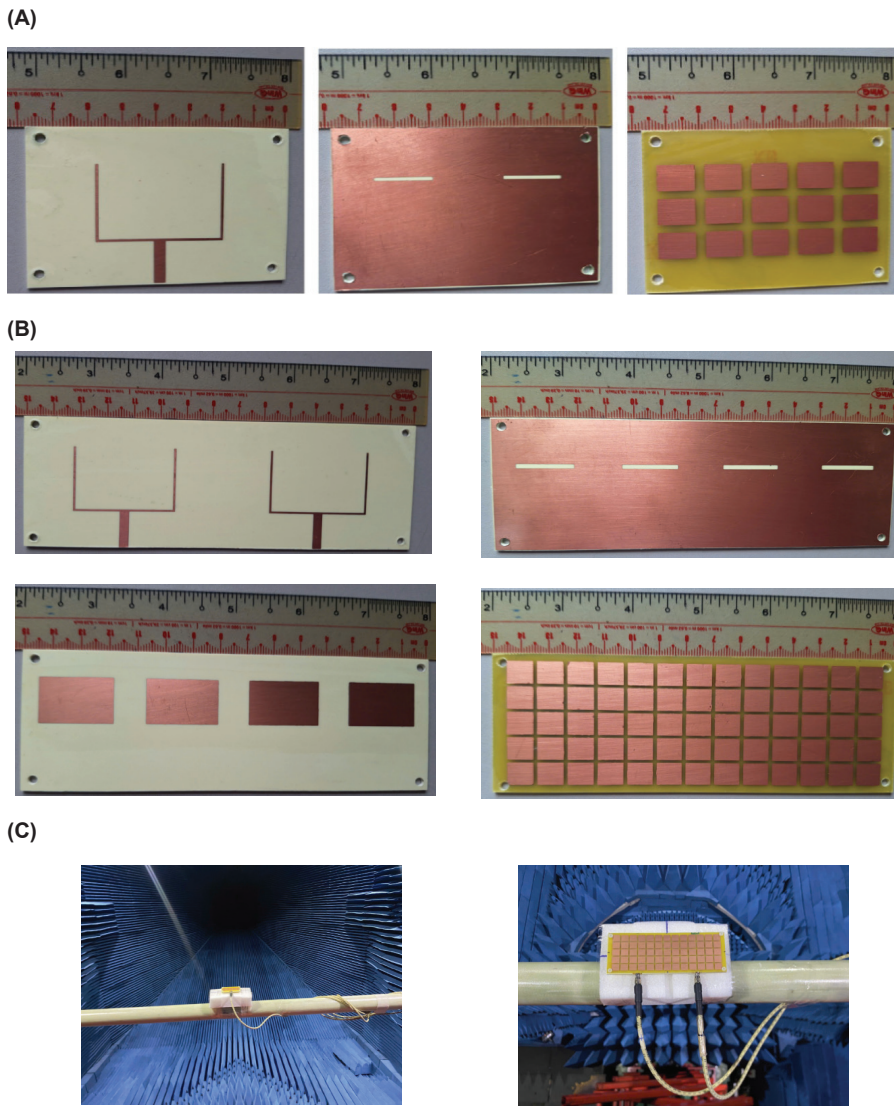


Fig. 11. The fabricated antenna prototypes. (A) Single antenna; (B) MIMO antenna; (C) Measurement setup. Source: Proposed by authors.

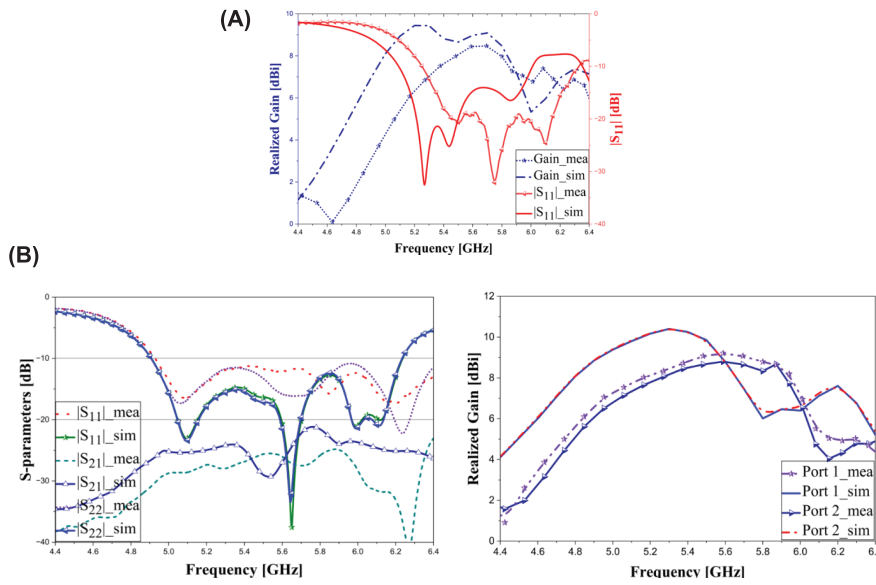


Fig. 12. Simulated and measured S-parameters and realised gain of: (A) The single antenna; (B) The MIMO antenna. Source: Proposed by authors.

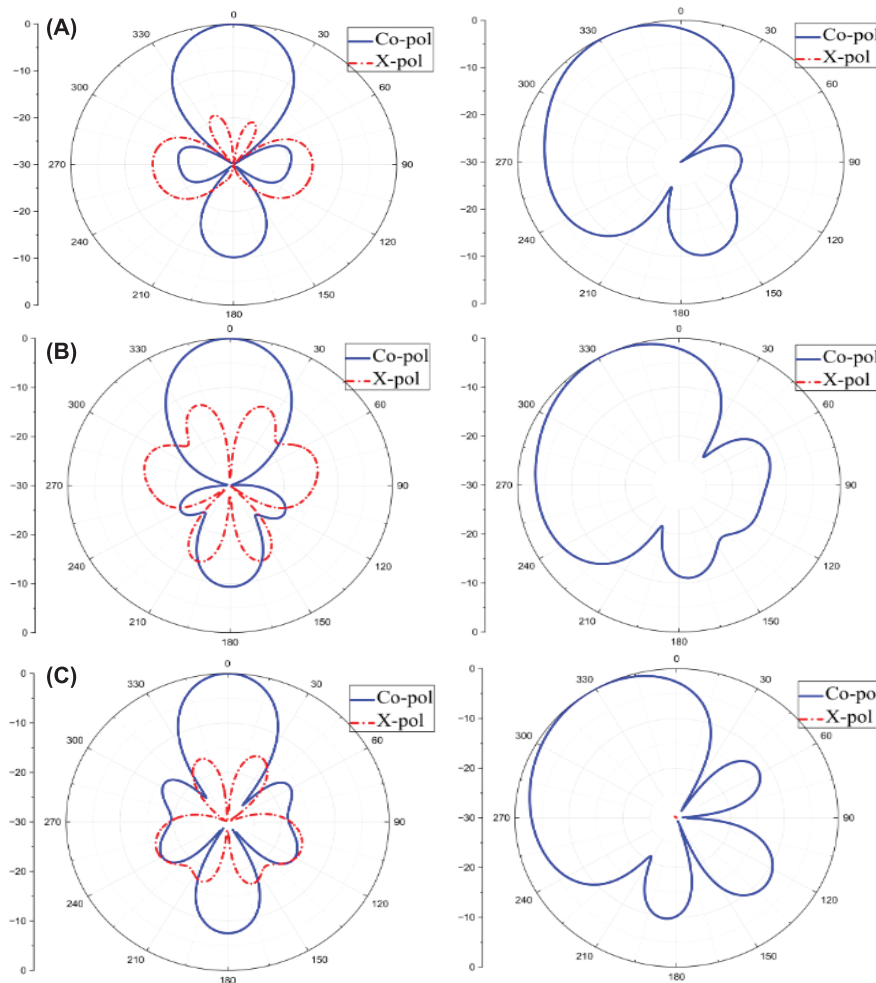


Fig. 13. Radiation patterns of the single antenna in the x-z (left) and y-z (right) planes: (A) 5.3 GHz, (B) 5.5 GHz, (C) 5.8 GHz. Source: Proposed by authors.

frequency range, the antenna gain ranges from 5.3 to 9.45 dBi for the simulation and 6 to 8.47 dBi for the measurement. For the MIMO antenna, the simulated and measured impedance bandwidths are found to be 25.3% (4.9-6.24 GHz) and 27.3% (4.9-6.4 GHz), respectively, while the gain values range between 6-10.4 dBi and 4.5-9.21 dBi.

Regarding the patterns (Figs. 13 and 14), the proposed antenna exhibits a broadside radiation pattern in the x-z plane. However, in the y-z plane, the main lobe deviates slightly from the broadside direction, which can be attributed to mutual coupling effects among the radiating elements. A noticeable discrepancy exists between the simulated and experimental results. These differences are primarily attributed to fabrication tolerances and the soldering process of the SMA connector, which results in a change in impedance. In addition, several secondary factors may have influenced the outcome, such as (i) imperfections in the measurement chamber, (ii) dielectric and copper losses in the substrate, and (iii) the presence of the RF feed cable inside the chamber during testing. Despite these discrepancies, the operational frequency range of the antenna is maintained, indicating that the observed differences remain within acceptable limits.

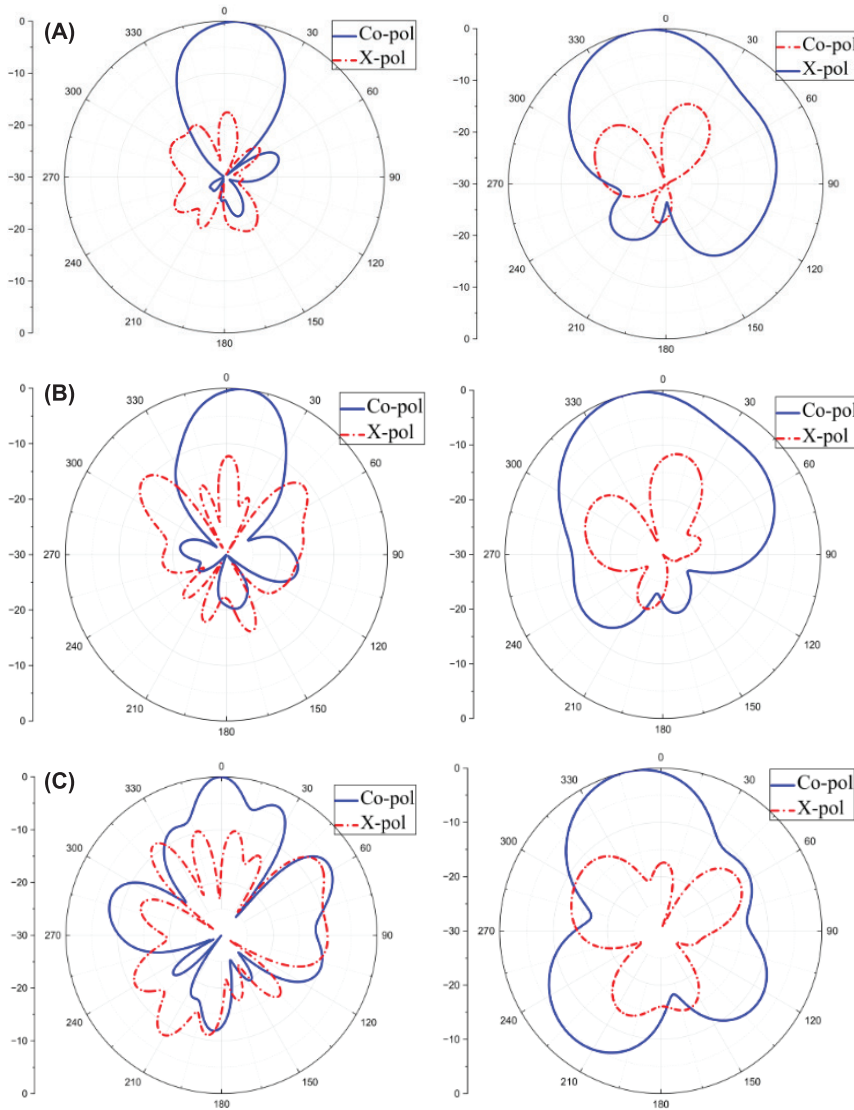


Fig. 14. The radiation pattern of the MIMO antenna in the x-z (left) and y-z (right) planes: (A) 5.3 GHz, (B) 5.5 GHz, (C) 5.8 GHz. Source: Proposed by authors.

5. Diversity property

The performance of MIMO antenna systems is strongly influenced by both pattern and polarisation diversity. To quantify this aspect, the Envelope Correlation Coefficient (ECC) is commonly employed, as it reflects the degree of interaction between antenna elements. ECC can be obtained from the scattering parameters, making it a convenient and widely adopted metric for antenna evaluation; it is expressed as follows:

$$ECC = \frac{|S_{11}^* S_{12} + S_{21}^* S_{22}|^2}{(1 - (|S_{11}|^2 + |S_{21}|^2))(1 - (|S_{22}|^2 + |S_{12}|^2))} \quad (1)$$

Alongside ECC, the Diversity Gain (DG) provides further insight into system performance by indicating how effectively diversity techniques enhance signal robustness. DG can be expressed as:

$$DG = 10\sqrt{1 - ECC^2} \quad (2)$$

Since DG is derived from ECC, it offers a straightforward way to interpret the impact of correlation on overall link quality.

For a more comprehensive assessment, additional parameters such as Channel Capacity Loss (CCL) and Total Active Reflection Coefficient (TARC) are also taken into account. These metrics, together with the coefficients, can be formulated using the expressions given in [27], helping to capture the effects of mutual coupling as well as the practical efficiency of the antenna system under real operating conditions.

Figure 15 illustrates the diversity performance of the proposed antenna system. Across the operating band of 4.9–6.3 GHz, the ECC remains below 0.002, while the DG is

maintained at approximately 10 dB. These results indicate a very low level of correlation between antenna elements, implying effective isolation and satisfactory diversity characteristics.

Furthermore, the TARC is consistently below -10 dB, and the CCL remains below 0.2 bits/s/Hz throughout the considered frequency range. It is worth noting that, for MIMO systems, a CCL value below 0.4 bits/s/Hz is generally regarded as acceptable [28]. Therefore, the results confirm that the proposed antenna meets the required criteria for reliable MIMO performance.

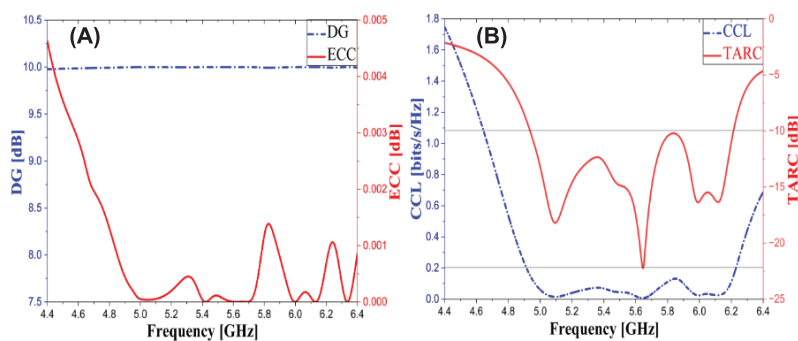


Fig. 15. The diversity performance of the proposed MIMO antenna: Diversity Gain (DG) and Envelope Correlation Coefficient (A); Channel Capacity Loss (CCL) and Total Active Reflection Coefficient (TARC) (B). Source: Proposed by authors.

6. Comparison and discussion

Finally, a comparison between the proposed antenna and related works is presented in Table 1. A review of existing designs reveals that although the designs in [11] and [17] achieve high efficiency, most antennas exhibit relatively narrow bandwidths, typically not exceeding 20%, with the exception of the configuration reported in [21]. Furthermore, several studies, including [5, 11, 17, 18, 21], are constrained by modest gain values, generally remaining below 9 dBi. In contrast, although the design in [4] achieves high gain, its larger physical size is a limiting factor. In addition, it can be observed that the ECC values of the designs in [5] and [18] are quite high, although they remain within acceptable standards (below 0.5) [29]. The proposed antenna was developed with the aim of striking a well-balanced trade-off among these key performance parameters, maintaining a low ECC (below 0.0015) and a reasonable radiation efficiency of 61% while achieving a significantly enhanced bandwidth (27.3%).

Table 1. Comparison of the proposed antenna and recent designs.

Ref.	Freq. (GHz)	Array Size	-10-dB BW (%)	Isolation (dB)	Peak Gain (dBi)	ECC	Radiation efficiency (%)	Size (λ^2)
[4]*	28	32	3	40	11.5	0.00005	58	$3.22 \times 3.08 \times \text{N/G}$
[5]	1.575	4	3.2	25	6	0.35	58.8	$\pi \times 0.328^2 \times 0.04$
[11]	5.9	4	4	28	6.5	0.001	90	$1.07 \times 1.07 \times 0.03$
[17]	8.34	2	10.1	18	8.8	N/G	90	N/G
[18]	3.4	4	17.96	20	5.03	0.02	50	$0.62 \times 0.62 \times \text{N/G}$
[21]	4.65	2	28	30	4	0.0002	77	$0.53 \times 0.53 \times 0.27$
This work	5.3	4	27.3	20	9.21	0.0015	61	$2.45 \times 0.9 \times 0.078$

* : the simulated results. Source: Proposed by authors.

7. Conclusions

In this article, a compact MIMO antenna for WLAN applications is proposed and experimentally verified at the frequency of 5.3 GHz. The proposed configuration, composed of two subarrays, shows that the operational bandwidth can be enhanced without increasing the complexity of the structure by introducing an FSS. It demonstrates that an FSS-based approach can be effective in overcoming the inherent bandwidth limitations of compact

MIMO antennas. The presence of FSS improves the impedance bandwidth from 2.3 to 27%. This demonstrates the effectiveness of the proposed approach to overcome the narrowband limitation in compact array configurations.

Furthermore, the antenna obtains the isolation higher than 20 dB in the operation frequency band and the maximum gain of 9.2 dBi, which is enough to decrease the mutual coupling effect and ensure the stable MIMO operation. Moreover, the proposed design achieves a balance among the crucial performance parameters such as bandwidth, gain, isolation and diversity characteristics, without the need of a complex configuration. The proposed antenna has the advantages of broadband and good isolation between the elements, and can be a good candidate for practical applications. The results confirm that good design methodology can result in substantial performance improvements, and indicate the importance of this study for the development of compact and high efficiency MIMO antenna systems.

CRediT author statement

Ngoc-Lan Nguyen: Conceptualisation and Methodology, Writing; Hien-Thuan Duong: Writing, Revision.

ACKNOWLEDGEMENTS

This article is supported by the Posts and Telecommunications Institute of Technology under grant number 02-2026-HV-VT2.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] A.J. Paulraj, D.A. Gore, R.U. Nabar, et al. (2004), "An overview of MIMO communications - A key to Gigabit Wireless", *Proc. IEEE*, **92(2)**, pp.198-218, DOI: 10.1109/JPROC.2003.821915.
- [2] F.A. Azhiri, R. Abdolee, B.M. Tazehkand (2019), "Effect of mutual coupling on the performance of STCM-MIMO Systems", *IEEE Wirel. Commun. Netw. Conf. Work*, pp.1-6, DOI: 10.1109/WCNCW.2019.8902766.
- [3] K.H. Chen, J. F. Kiang (2016), "Effect of mutual coupling on the channel capacity of MIMO systems", *IEEE Trans. Veh. Technol.*, **65(1)**, pp.398-403, DOI: 10.1109/TVT.2015.2397033.
- [4] S. Aourik, A. Errkik, A. Oukaira, et al. (2023), "An advanced array configuration antenna based on mutual coupling reduction", *Electron.*, **12(7)**, DOI: 10.3390/electronics12071707.
- [5] A. Madni, W.T. Khan (2023), "Design of a compact 4-Element GNSS antenna array with high isolation using a defected ground structure (DGS) and a microwave absorber", *IEEE Open J. Antennas Propag.*, **4**, pp.779-791, DOI: 10.1109/OJAP.2023.3298773.
- [6] I.U. Din, S. Ullah, N. Mufti, et al. (2023), "Metamaterial-based highly isolated MIMO antenna system for 5G smartphone application", *Int. J. Commun. Syst.*, **36(3)**, DOI: 10.1002/dac.5392.
- [7] J. Kim, J. Ju, S. Eom, et al. (2013), "Four-channel MIMO antenna for WLAN using hybrid structure", *Electron. Lett.*, **49(14)**, pp.875-876, DOI: 10.1049/el.2013.0994.
- [8] M. Elahi, S. Koziel, L. Leifsson (2025), "Isolation improvement in MIMO antenna with a simple hybrid technique of orthogonal and inverse currents", *AEU - Int. J. Electron. Commun.*, **188**, DOI: 10.1016/j.aeue.2024.155576.
- [9] C.K. Ghosh, M. Pratap, R. Kumar (2020), "Mutual coupling reduction of microstrip mimo antenna using microstrip resonator", *Wirel. Pers. Commun.*, **112(3)**, pp.2047-2056, DOI: 10.1007/s11277-020-07138-z.
- [10] S. Hamdan, E.K.I. Hamad, H.A. Mohamed, et al. (2024), "High-performance MTM inspired two-port MIMO antenna structure for 5G/IoT applications", *J. Electr. Eng.*, **75(3)**, pp.214-223, DOI: 10.2478/jee-2024-0026.
- [11] S.N.R. Rizvi, M.A. Sujian, N. Hussain, et al. (2024), "Isolation enhancement of a capacitively-fed MIMO antenna using a quasi-fractal parasitic element and defected ground structure", *Heliyon*, **10(20)**, DOI: 10.1016/j.heliyon.2024.e39228.
- [12] K.C. Ravi, J. Kumar (2022), "Miniaturized parasitic loaded high-isolation MIMO Antenna for 5G applications", *Sensors*, **22(19)**, DOI: 10.3390/s22197283.
- [13] A. Li, S. Singh, D. Sievenpiper (2018), "Metasurfaces and their applications", *Nanophotonics*, **7(6)**, pp.989-1011, DOI: 10.1515/nanoph-2017-0120.
- [14] A. Ali, A. Mitra (2022), "Metamaterials, Metasurfaces: A Review from the Perspectives of Materials, Mechanisms and Advanced Metadevices", *Nanomaterials*, **8(9)**, DOI: 10.3390/nano12061027.
- [15] R.S. Anwar, L. Mao, H. Ning (2018), "Frequency selective surfaces: A review", *Appl. Sci.*, **8(9)**, pp.1-46, DOI: 10.3390/app8091689.
- [16] F. Guidoum, M.L. Tounsi, T.P. Vuong (2021), "Enhancing 5G antenna performance by using 3D FSS structures", *Int. J. RF Microw. Comput. Eng.*, **31(8)**, DOI: 10.1002/mmce.22739.
- [17] S.U. Rahman (2021), "Angularly stable frequency selective surface for the gain enhancement of isolated multiple input multiple output antenna", *Microw. Opt. Technol. Lett.*, **63(11)**, pp.2803-2810, DOI: 10.1002/mop.32980.
- [18] A. Desai, H.T. Hsu, Y.F. Tsao (2024), "FSS based high gain optically transparent MIMO antenna for Sub-6 GHz 5G mid-band applications", *Optik*, **307**, DOI: 10.1016/j.ijleo.2024.171829.
- [19] A.B. Devarapalli, T. Moyra (2023), "Design of a metamaterial loaded W-shaped patch antenna with FSS for improved bandwidth and gain", *Silicon*, **15(4)**, pp.2011-2024, DOI: 10.1007/s12633-022-02123-6.
- [20] A.K. Tiwari, A. Ahmed, M. Bharti (2025), "A wide-band frequency-selective surface for enhancing the performance of microstrip antenna", *Int. J. Microw. Wirel. Technol.*, **17(1)**, pp.132-139, DOI: 10.1017/S1759078725000212.
- [21] R. Mondal, P.S. Reddy, D.C. Sarkar, et al. (2021), "Investigation on MIMO antenna for very low ECC and isolation characteristics using FSS and metal-wall", *AEU - Int. J. Electron. Commun.*, **135**, DOI: 10.1016/j.aeue.2021.153708.
- [22] L. Si, H. Jiang, X. Lv, et al. (2019), "Broadband extremely close-spaced 5G MIMO antenna with mutual coupling reduction using metamaterial-inspired superstrate", *Opt. Express*, **27(3)**, DOI: 10.1364/oe.27.003472.
- [23] N.L. Nguyen, C.D. Bui, Q.S. Nguyen (2024), "Design of a compact antenna with broadband and high gain", *Electromagnetics*, **44(1)**, pp.18-31, DOI: 10.1080/02726343.2023.2300840.
- [24] G. Pachaiyappan, P. Ramanujam (2021), "Design of low profile co-axial fed high gain stacked patch antenna for Wi-Fi/WLAN/Wi-Max applications", *Frequenz*, **75(1-2)**, pp.27-34, DOI: 10.1515/freq-2020-0021.
- [25] S.C. Gao, L.W. Li, M.S. Leong (2001), "A novel dual-polarized, wide-band microstrip patch antenna with aperture coupling", *IEEE Antennas and Propagation Society International Symposium*, **4**, pp.78-81, DOI: 10.1109/aps.2001.959404.
- [26] Z. Li, X. Weng, X. Yi, et al. (2024), "A broadband second-order bandpass frequency selective surface for microwave and millimeter wave application", *Sci. Rep.*, **14(1)**, pp.1-10, DOI: 10.1038/s41598-024-62228-3.
- [27] N. Raha (2022), "Design and simulation of compact MIMO antenna for the 5G communication in C-Band", *J. Phys. Conf. Ser.*, **2312(1)**, DOI: 10.1088/1742-6596/2312/1/012032.
- [28] Y.K. Choukiker, S.K. Sharma, S.K. Behera (2014), "Hybrid fractal shape planar monopole antenna covering multiband wireless communications with MIMO implementation for handheld mobile devices", *IEEE Antennas and Propag.*, **62(3)**, pp.1483-1488, DOI: 10.1109/TAP.2013.2295213.
- [29] N. Hussain, T.D. Pham, H.H. Tran (2022), "Circularly polarized MIMO antenna with wideband and high isolation characteristics for C-Band communication systems", *Micromachines*, **13(11)**, pp.1-14, DOI: 10.3390/mi13111894.

Zero trust policy enforcement for OSPF link-state advertisements via blockchain-based smart contracts

Chinh Huynh Nguyen^{1*}, Tam Thanh Huynh², Doan Cong Dinh¹

¹Ho Chi Minh City University of Technology and Engineering, 1 Vo Van Ngan Street, Thu Duc Ward, Ho Chi Minh City, Vietnam

²Ho Chi Minh City University of Industry and Trade, 140 Le Trong Tan Street, Tay Thanh Ward, Ho Chi Minh City, Vietnam

Received 23 March 2026; revised 24 July 2026; accepted 15 August 2026

Abstract:

The Open Shortest Path First (OSPF) protocol remains vulnerable to insider control-plane attacks in which compromised but authenticated routers inject unauthorised link-state advertisements (LSAs). Existing defences mainly detect or mitigate malicious LSAs after propagation and rarely verify compliance with routing policies beforehand. This study adopts a zero trust approach requiring continuous policy-driven verification of all LSAs regardless of sender authentication. We propose a framework that validates each LSA against administrator-defined rules, including router ID, network prefix, LSA type, and flooding area, using smart contracts on a private Ethereum blockchain. This ensures decentralised, tamper-resistant, and auditable policy enforcement without relying on a central authority. The framework includes Policy Initialisation, Policy Update, and Policy-based LSA Verification, enabling proactive authorisation without modifying OSPF. Security analysis shows mitigation of insider attacks such as LSA falsification, adjacency-based injection, single-path injection, and link-state database poisoning. Experiments on a multi-area OSPF topology with 15 routers across four areas confirm correct policy enforcement across eight evaluation scenarios covering prefix injection, type manipulation, cross-area flooding, and role spoofing attacks. Verification latency averages 162.86 ms with a 95th percentile of 216.42 ms, well within OSPF operational timing constraints, while incurring zero runtime gas cost. Scalability evaluation demonstrates stable verification latency (150-165 ms mean, P95 204 ms) from 100 to 10,000 policy entries, confirming O(1) lookup complexity.

Keywords: blockchain-based policy enforcement, OSPF routing security, zero trust architecture.

1. Introduction

Interior Gateway Protocols (IGPs) constitute the control-plane foundation of modern IP networks by enabling routers within an autonomous system (AS) to exchange topology information and compute forwarding paths. Among existing IGPs, Open Shortest Path First (OSPF) is widely deployed in enterprise, service provider, and critical infrastructure networks due to its scalability, fast convergence, and standardised link-state architecture [1, 2]. In OSPF, routers disseminate network topology information through Link-State Advertisements (LSAs), allowing each router to construct a global link-state database (LSDB) and independently compute shortest paths using the Shortest Path First (SPF) algorithm.

OSPF version 2 (OSPFv2), designed for IPv4 networks, includes built-in authentication mechanisms such as plaintext passwords, MD5 authentication, and later extensions supporting HMAC-SHA [3]. These mechanisms ensure that routing messages originate from authenticated neighbours and have not been modified in transit. However, OSPFv2 relies on shared symmetric keys, meaning that compromise of a single router or authentication key can enable impersonation, message replay, and the injection of falsified routing information into the control plane [4, 5]. In addition, its reliance on the Address Resolution Protocol (ARP) for neighbour discovery introduces further attack surfaces, including ARP poisoning and man-in-the-middle attacks [6].

*Corresponding author: Email: chinhhn@hcmute.edu.vn

OSPF version 3 (OSPFv3) addresses several architectural limitations of its predecessor by introducing native IPv6 support, separating topology information from IP addressing, employing link-local addresses for neighbour adjacencies, and mandating IPsec to secure OSPF packet exchanges [7, 8]. While these improvements strengthen the protection of routing message transport, OSPFv3 fundamentally preserves the same trust-based routing model. IPsec ensures the confidentiality and integrity of communication channels but does not evaluate the semantic correctness or administrative authorisation of routing information advertised by authenticated routers.

Consequently, both OSPF versions remain vulnerable to a class of control-plane attacks originating from compromised internal routers. An attacker who gains control of an authenticated router may inject falsified LSAs into the LSDB, corrupting other routers' topology views and causing routing loops, traffic blackholing, or persistent traffic diversion [9-12]. These attacks exploit a fundamental design assumption: OSPF validates the syntactic structure and temporal freshness of LSAs but does not verify whether the advertising router is administratively authorised to announce a given prefix, originate a specific LSA type, or flood an advertisement within a particular area. Because such manipulations remain protocol-compliant, they can evade conventional monitoring approaches that rely on traffic anomalies in the data plane.

A broad range of approaches has been proposed to address OSPF routing security. Detection-based techniques employ statistical analysis, anomaly detection, and runtime monitoring to identify abnormal routing behaviour after malicious advertisements have been disseminated [13-17]. Other efforts focus on vulnerability analysis tools [18], trust-aware routing strategies [19-21], cryptographic enhancements and formal verification [22-24], and policy-based verification mechanisms [25-27]. Despite these advances, most existing solutions remain reactive, operating after policy-violating LSAs have already propagated, or function at the routing and forwarding layers rather than preventing unauthorised advertisements from entering the LSDB.

To address this limitation, this paper adopts a zero trust paradigm [28] that eliminates implicit trust among routers and requires continuous

verification of control-plane advertisements, even when they originate from authenticated devices. Under this model, no router is inherently trusted to advertise any routing information; each LSA must be explicitly authorised against administrator-defined policies before it can influence the LSDB.

Enforcing such policies consistently across a distributed routing domain presents a design challenge. A centralised policy server would introduce a single point of failure and trust, which is particularly problematic under the insider threat model where any network device may be compromised. To avoid this vulnerability, we employ blockchain technology as the policy storage and enforcement substrate [29, 30]. Routing authorisation policies stored on a private blockchain and enforced through smart contracts provide decentralised, tamper-resistant, and auditable verification without relying on a centralised authority. Each verification module independently queries the blockchain to validate LSA attributes, ensuring that policy enforcement remains consistent even if individual network components are compromised.

Based on this design, we propose a zero trust-based security framework for policy-driven authorisation of OSPF LSAs. The framework verifies four key attributes, namely the advertising router identifier, network prefix, LSA type, and flooding area, against authorisation policies stored on the blockchain. The approach operates transparently and requires no modification to existing OSPF implementations.

The main contributions of this paper are as follows:

- (1) We propose a zero trust-based security framework for OSPF that replaces implicit router trust with continuous, policy-driven verification of LSAs at the point of reception, explicitly addressing insider threats in the routing control plane.
- (2) We design a blockchain-assisted policy enforcement layer in which administrator-defined authorisation rules are stored as smart contracts, providing decentralised and tamper-resistant policy management without introducing a single point of trust or failure.
- (3) We formalise three operational schemes, Policy Initialisation, Policy Update, and Policy-based LSA Verification, to enable proactive authorisation of routing advertisements while remaining fully transparent to OSPF operations.

(4) We implement and evaluate the proposed framework in an emulated multi-area OSPF environment, demonstrating correct policy enforcement across eight evaluation scenarios with an average verification latency of 162.86 ms and zero runtime on-chain gas cost.

2. Background

2.1. OSPF routing and link-state advertisements

OSPF disseminates topology information through LSAs, which are flooded within logical routing domains called areas and stored in a synchronised LSDB. Each router independently computes forwarding paths from the LSDB using the SPF algorithm [31].

Each LSA comprises a fixed header and type-specific topology data. The header includes fields such as LSA type, advertising router identifier, link-state ID, sequence number, age, and checksum, which allow routers to maintain LSA freshness, detect outdated entries, and ensure reliable propagation. The body encodes topology information including network prefixes, link types, and metrics. Four LSA types are central to topology construction:

- Router LSAs (Type 1) describe local interfaces and connected prefixes within an area.
- Network LSAs (Type 2) represent multi-access segments and are originated by the Designated Router.
- Summary LSAs (Types 3 and 4) are generated by Area Border Routers (ABRs) to advertise inter-area reachability.
- AS-External LSAs (Type 5) are originated by Autonomous System Boundary Routers (ASBRs) to announce external routes.

From a security perspective, four LSA attributes constitute the primary authorisation surface exploitable in insider-driven attacks: the advertising router identifier, the advertised network prefix, the LSA type, and the flooding area. Standard OSPF validates these attributes only for protocol consistency and syntactic correctness. It does not verify whether a router is administratively authorised to advertise a given prefix, originate a specific LSA type, or flood an advertisement within a particular area. This gap allows a compromised router to inject policy-violating LSAs that remain protocol-compliant while corrupting the LSDB and manipulating routing decisions network-wide.

2.2. Zero trust architecture

The zero trust security paradigm eliminates implicit trust granted through network location or prior authentication, instead requiring continuous and explicit verification of every entity and action according to predefined policies [28]. Under this model, no participant is inherently trusted; each operation must be authorised at the time it occurs, regardless of authentication status.

Applied to routing control planes, zero trust implies that each LSA must be verified against authorisation policies at the point of reception. Rather than assuming that an authenticated router will only advertise legitimate routing information, the verification process treats every advertisement as potentially unauthorised until it is validated against the policy store. This shift from perimeter-based trust to per-transaction verification directly addresses the insider threat model in which compromised routers abuse their authenticated status to inject malicious LSAs.

2.3. Blockchain-based policy enforcement

Enforcing routing authorisation policies across a distributed domain requires a policy store that is consistent, tamper-resistant, and resilient to single points of failure. A centralised policy server can provide consistency but introduces a single point of trust and availability, which is problematic under the insider threat model where any network component may be compromised.

Blockchain technology addresses these requirements by providing a decentralised and immutable ledger for policy storage. Authorisation policies encoded as smart contracts are replicated across all blockchain nodes, ensuring that every verification module queries the same policy state without depending on a single authority [29, 30]. Smart contracts execute deterministically, producing consistent verification results regardless of which node processes the query. This property guarantees uniform policy enforcement across the routing domain.

Combining zero trust principles with blockchain-based policy management enables proactive, per-LSA authorisation at the control-plane level. Policy-violating LSAs are identified and mitigated at the point of reception, before they can influence the LSDB, regardless of the authentication status of the advertising router.

3. Related work

The security of the OSPF protocol has been extensively studied due to the critical role of routing control planes in network stability. This section reviews existing approaches across five categories and positions the proposed framework relative to the state of the art.

OSPF vulnerability analysis and attack demonstrations: Early research focused on identifying structural vulnerabilities in OSPF and demonstrating the feasibility of control-plane manipulation attacks. G. Nakibly, et al. (2012) [9], G. Nakibly, et al. (2014) [10] showed that deterministic protocol parameters and fight-back mechanisms can be exploited to inject persistent falsified LSAs. Y. Song, et al. (2017) [12] demonstrated novel attacks that exploit adjacency formation and flooding procedures to maintain prolonged influence over routing decisions. E. Kaffashi, et al. (2014) [11] further expanded the threat landscape by demonstrating structural LSDB poisoning attacks. These studies collectively establish that OSPF's lack of semantic LSA validation constitutes a fundamental security gap.

Detection-based approaches: A substantial body of work addresses OSPF security through anomaly detection. B. Al-Musawi, et al. (2020) [13], B. Al-Musawi, et al. (2018) [15] applied non-linear dynamics and recurrence quantification analysis to detect abnormal LSA dissemination patterns. S. Jin, et al. (2019) [16] proposed health-status analysis for identifying anomalous behaviour in core routers. N. Devir, et al. (2019) [14] introduced topology-agnostic runtime monitoring that identifies routing anomalies without prior knowledge of the network structure, while S. Sengupta, et al. (2022) [17] developed continuous in-network monitoring techniques. These approaches are effective at identifying suspicious behaviour; however, they are inherently reactive, operating after malicious LSAs have already propagated through the network and influenced routing decisions. Consequently, they cannot prevent the initial corruption of the LSDB.

Vulnerability assessment tools: P. Kasemsuwan, et al. (2017) [18] developed the OSPF Vulnerability Scanner (OSV), a systematic tool for identifying configuration weaknesses and protocol misconfigurations in operational OSPF environments. While OSV improves visibility into potential security risks, it functions primarily as an offline diagnostic tool and does not provide runtime enforcement.

Trust-aware and resilience-oriented routing: Several approaches attempt to mitigate the influence of compromised routers by incorporating trust metrics into routing decisions. K. Obelovska, et al. (2023) [19], K. Obelovska, et al. (2024) [20] proposed methods for routing around untrusted routers by adjusting path selection algorithms and minimising the number of untrusted nodes on forwarding paths. These approaches improve network robustness at the forwarding layer but do not validate the authorisation of individual LSAs at the control-plane level. A compromised router's falsified advertisements may still propagate and corrupt the LSDB even if its forwarding influence is limited.

Cryptographic enhancements and policy-based verification: S. Othmen, et al. (2024) [22] proposed group key management mechanisms for secure routing in IoT systems. J. Pereira, et al. (2025) [23] applied formal verification to next-generation internet router implementations, and G.V. Lakshmi, et al. (2022) [24] designed security frameworks for trusted routing in MANETs. X. Shao, et al. (2020) [25] developed policy verification techniques for large-scale inter-domain routing. Y. Bai, et al. (2022) [26] investigated reinforcement learning-based routing optimisation, while A. Kumar, et al. (2025) [27] proposed secure source-routing architectures with cryptographic path enforcement. These efforts strengthen various aspects of routing security, including message integrity, policy compliance, and path assurance. However, they generally assume that control-plane information originating from authenticated routers is trustworthy, and therefore do not address the specific threat of insider-driven LSA manipulation in OSPF.

Summary and positioning. Table 1 compares existing approaches across four dimensions: (i) proactive prevention of malicious LSA propagation, (ii) per-LSA semantic authorisation, (iii) transparency to the OSPF protocol, and (iv) decentralised policy enforcement.

Table 1. Comparison of existing OSPF security approaches.

Approach	Proactivity	Per-LSA	Transparency	Decentralisation
[13-17]	No	No	Yes	N/A
[18]	No	No	Yes	N/A
[19, 20]	No	No	No	N/A
[22-24]	Partial	No	No	No
[25-27]	Partial	Partial	Partial	No
This work	Yes	Yes	Yes	Yes

Source: Compiled by the authors based on [13-20] and [22-27].

As summarised in Table 1, existing approaches typically address only subsets of the considered properties, reflecting different security objectives and deployment assumptions. Detection-based methods lack proactive prevention. Trust-aware routing operates at the forwarding layer rather than the control plane. Cryptographic enhancements verify message authenticity but not routing authorisation. The proposed framework addresses this gap by enforcing proactive, per-LSA semantic authorisation at the point of reception through blockchain-based smart contracts, without requiring modification to the OSPF protocol.

4. Proposed solution

4.1. System model

To enable policy-driven authorisation of OSPF control-plane advertisements, the proposed architecture introduces an external verification layer integrated with blockchain-based policy management, as illustrated in Fig. 1. The system comprises five components.

Administrator: The administrator is responsible for defining and updating routing authorisation policies through the smart contract deployed on the blockchain.

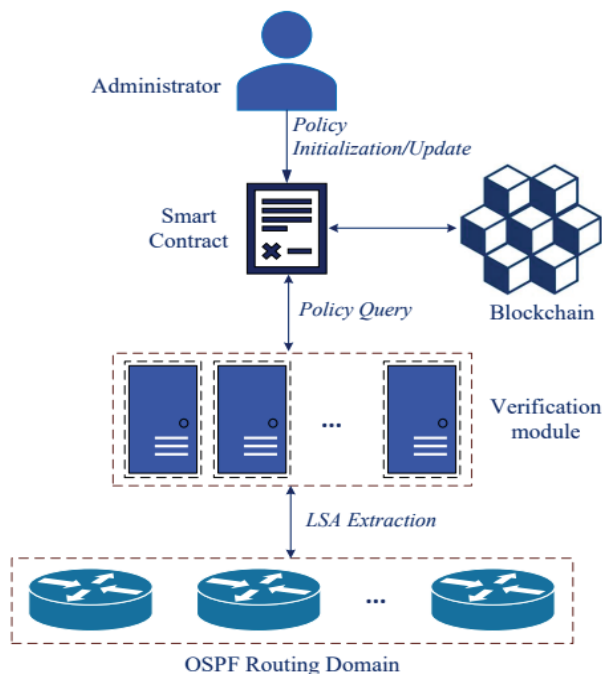


Fig. 1. The system model. Source: Authors' own work.

OSPF routing domain: The routing domain consists of routers operating under the standard OSPF protocol. The proposed framework preserves

existing authentication mechanisms and does not modify OSPF message formats or routing operations, thereby maintaining full interoperability with existing deployments.

Verification module layer: Each verification module serves as the zero trust enforcement point for a local OSPF router. Upon receiving an LSA, the module extracts a policy-relevant attribute tuple (*RID*, *Net*, *AreaID*, *Type*) and verifies it against authorisation policies enforced through the blockchain-based smart contract. If a policy violation is detected, the module triggers configurable containment actions to mitigate the propagation or impact of the unauthorised advertisement.

Smart contract: The *OSPFPolicy* smart contract encodes routing authorisation rules and access control policies. It exposes three core functions, *PolInitiation*, *PolUpdate*, and *PolVerification*, to provide deterministic and transparent validation of LSA attributes while restricting policy management to the authorised administrator.

Blockchain network: A private Ethereum blockchain serves as the decentralised and tamper-resistant storage layer for routing policies. The immutable ledger of policy states ensures consistent enforcement across all verification modules and provides a complete audit trail of policy changes.

4.2. Threat model

The threat model explicitly distinguishes between trusted infrastructure components and potentially compromised routing devices.

4.2.1. Trusted components

Three categories of components are assumed trustworthy:

Administrator: The administrator is the sole entity authorised to deploy or update routing policies. This assumption is standard in policy-based security architectures, as the administrator defines the ground truth for authorised routing behaviour.

Blockchain infrastructure: The smart contract and the underlying private blockchain are assumed to operate correctly. Because the blockchain is deployed as a private, permissioned network controlled by the network operator, the risk of external compromise is substantially lower than in public blockchain deployments. The blockchain guarantees integrity, consistency, and auditability of routing policies.

Verification modules: These modules extract LSA attributes and query the smart contract for policy decisions. They are deployed on dedicated infrastructure under the administrator's control, analogous to network management stations or SDN controllers in conventional architectures. Their trustworthiness derives from their isolation from the OSPF data plane and their limited, well-defined functionality.

We acknowledge that the trust placed in these components is a design trade-off. The alternative, trusting no component, would make policy enforcement impossible. The proposed model minimises the trusted computing base to components that are (i) under direct administrative control, (ii) not exposed to the OSPF data plane, and (iii) independently verifiable through blockchain auditability.

Zero trust architectures still require a minimal trusted computing base for policy definition and enforcement. In our design, trust is minimised to administratively controlled components outside the OSPF forwarding plane.

4.2.2. Adversary capabilities

Within the OSPF routing domain, all routers are considered potentially compromised. We assume a strong adversarial model in which an attacker may fully compromise an authenticated router and obtain valid OSPF credentials. The attacker can therefore:

- Legitimately participate in OSPF message exchanges and neighbour adjacency formation.
- Generate syntactically valid LSAs with manipulated attributes, including falsified router identifiers, unauthorised network prefixes, incorrect LSA types, or invalid flooding area assignments.

The attacker's objective is to corrupt the LSDBs of other routers, leading to incorrect topology views, routing loops, traffic diversion, blackholing, or persistent routing instability.

4.2.3. Out-of-scope attacks

The following attack vectors are outside the scope of this work:

Management channel attacks: Verification modules communicate with routers through secure, authenticated management interfaces. Attacks targeting these channels (e.g., denial-of-service, command interception) are not addressed.

Dynamic LSA attribute manipulation: The current framework validates static policy attributes (*RID*,

Net, *AreaID*, *Type*) but does not monitor dynamic LSA fields such as sequence numbers, LSA age, or refresh timing. Attacks exploiting abnormal manipulation of these dynamic properties require stateful temporal analysis and are deferred to future work.

Blockchain consensus attacks: As a private, permissioned network, the blockchain is assumed resilient to consensus-level attacks (e.g., 51% attacks) that are primarily concerns in public blockchain environments.

4.3. System setup

4.3.1. OSPF routing domain configuration

The OSPF routing domain consists of routers configured using standard OSPF mechanisms. Each router is assigned a unique router identifier (*RID*). The network administrator defines the set of IP network prefixes and associates each prefix with a corresponding OSPF area. Let *Net* denote an advertised network prefix and *AreaID* denote the OSPF area identifier in which the prefix is originated or flooded.

During normal operation, routers exchange LSAs describing local connectivity. Each LSA is characterised by the attribute tuple (*RID*, *Net*, *AreaID*, *Type*), where *Type* specifies the semantic role of the advertisement (e.g., Router LSA, Network LSA, Summary LSA, or AS-External LSA). Among these attributes, *RID*, *Net*, and *Type* are extracted directly from the LSA payload, while *AreaID* is inferred from accompanying protocol metadata, since the flooding scope is not explicitly encoded in the LSA body.

Routers provide received LSAs together with protocol metadata to the local verification module. When a policy violation is detected, containment actions are issued through secure management interfaces. These actions may include suppressing route installation, applying prefix filtering rules, or restricting the advertising router's influence on routing decisions. All enforcement operates externally to the OSPF protocol, preserving protocol transparency.

4.3.2. Verification modules

Each verification module validates received LSAs before they influence the LSDB. The extraction procedure (Algorithm 1) derives the tuple (*RID*, *Net*, *AreaID*, *Type*) from the LSA payload and its associated metadata. This tuple is submitted to the *PolVerification* function (Algorithm 4) on the smart contract. If the result is *True*, the LSA proceeds through normal OSPF processing. Otherwise, the module initiates containment actions.

Algorithm 1: DataExtracting

Input: LSA, Metadata
Output: RID, Net, AreaID, Type

```

1 (RID, Net, Type) ← extract(LSA);
2 AreaID ← inferArea(Metadata);
3 return (RID, Net, AreaID, Type);

```

4.3.3. Smart contract and blockchain

Cryptographic identities authenticate all interactions with the smart contract. The administrator holds a key pair (PU_{admin}, PR_{admin}) and each verification module holds a key pair (PU_{vm}, PR_{vm}) . Blockchain addresses are derived from public keys, and transaction authenticity is verified through digital signatures.

The *OSPFPolicy* smart contract maintains two data structures:

- *VMList*: The set of blockchain addresses of authorised verification modules permitted to invoke the *PolVerification* function.

- *RouterTable*: A mapping from router identifiers to sets of authorised routing attribute tuples (*Net*, *AreaID*, *Type*). Each entry defines the permitted routing advertisements for a given *RID*.

The contract exposes three functions:

PolInitiation (Algorithm 2). Registers a new routing policy entry (*RID*, *Net*, *AreaID*, *Type*) in *RouterTable*. Access is restricted to the administrator. The function rejects duplicate entries, sets the policy state to *active*, and records a timestamp for auditing.

Algorithm 2: PolInitiation

Input: RID, Net, AreaID, Type
Output: Null

```

1 if Caller ≠ Administrator then
2   Revert
3 End
4 if RouterTable[RID][Net][AreaID][Type] exists then
5   Revert
6 End
7 Policy.RID ← RID;
8 Policy.Net ← Net;
9 Policy.AreaID ← AreaID;
10 Policy.Type ← Type;
11 Policy.State ← True;
12 Policy.Timestamp ← current time;
13 RouterTable[RID][Net][AreaID][Type] ← Policy;

```

PolUpdate (Algorithm 3). Modifies the state of an existing policy entry to either *active* or *inactive*. The function verifies administrator authorisation, confirms that the target entry exists, updates the policy state, and refreshes the timestamp.

Algorithm 3: PolUpdate

Input: RID, Net, AreaID, Type, State
Output: Null

```

1 if Caller ≠ Administrator then
2   revert
3 end
4 if RouterTable[RID][Net][AreaID][Type] does not exist then
5   revert
6 end
7 Policy ← RouterTable[RID][Net][AreaID][Type];
8 Policy.State ← State;
9 Policy.Timestamp ← current time;
10 RouterTable[RID][Net][AreaID][Type] ← Policy;

```

PolVerification (Algorithm 4). Evaluates a received LSA's attribute tuple against *RouterTable*. The function verifies that the caller is an authorised verification module listed in *VMList*, checks for a matching active policy entry, and returns *True* if the LSA is authorised or *False* otherwise. Implemented as a Solidity view function, the *PolVerification* function does not modify blockchain state and incurs zero gas cost when invoked off-chain.

Algorithm 4: PolVerification

Input: RID, Net, AreaID, Type
Output: bool

```

1 if Caller ∉ VMList then
2   return False
3 end
4 if RouterTable[RID][Net][AreaID][Type] does not exist then
5   return False
6 end
7 Policy ← RouterTable[RID][Net][AreaID][Type];
8 if Policy.State ≠ active then
9   return False
10 end
11 return True

```

4.4. Operation schemes

This section describes the three operational procedures that coordinate interactions among the administrator, verification modules, and the smart contract.

4.4.1. Policy initialisation

The policy initialisation is executed by the administrator to register authorisation policies for all routers in the OSPF domain.

Step 1. The administrator collects routing configuration parameters from the OSPF domain and organises them as policy tuples (RID , Net , $AreaID$, $Type$).

Step 2. For each tuple, the administrator generates a blockchain transaction invoking *PolInitiation* (Algorithm 2), signs it with PR_{admin} , and propagates it to the blockchain network.

Step 3. The smart contract verifies the caller's authorisation. If the caller is the administrator and the entry does not already exist, the policy is recorded in *RouterTable*.

4.4.2. Policy update

The policy update allows the administrator to activate or deactivate existing policies in response to network topology changes or administrative decisions.

Step 1. The administrator identifies the target policy entry (RID , Net , $AreaID$, $Type$) and the desired state *State* {*active*, *inactive*}.

Step 2. A signed transaction invoking *PolUpdate* (Algorithm 3) is submitted to the blockchain.

Step 3. The smart contract verifies authorisation, confirms the entry exists, updates the state, and refreshes the timestamp.

4.4.3. Policy-based LSA verification

The LSA verification is the runtime procedure executed by verification modules for every received routing advertisement.

Step 1. The verification module receives an LSA and its associated protocol metadata from the local OSPF router.

Step 2. The module extracts the attribute tuple (RID , Net , $AreaID$, $Type$) using Algorithm 1.

Step 3. The module invokes the *PolVerification* function (Algorithm 4) on the smart contract. If the result is *True*, normal OSPF processing continues. If *False*, the module triggers containment actions, which may include suppressing route installation, restricting LSA propagation, or isolating the advertising router's influence on routing decisions.

5. Security analysis

This section analyses the security properties of the proposed framework under the threat model defined in section 4.2. We evaluate how policy-based verification mitigates four representative OSPF control-plane attacks initiated by authenticated but compromised routers.

The core enforcement mechanism is semantic authorisation: for every received LSA, the verification module extracts the attribute tuple (RID , Net , $AreaID$, $Type$) and evaluates it against the active routing policies stored in *RouterTable* on the blockchain. Any LSA whose attributes do not match an active policy entry is classified as a policy violation, and containment actions are triggered. We analyse four representative attack categories below.

5.1. LSA falsification

Attack description: In LSA falsification attacks, a compromised router injects forged LSAs containing unauthorised prefixes, fabricated topology information, or spoofed router identifiers into the LSDB [10]. Because OSPF floods LSAs domain-wide, falsified advertisements can rapidly corrupt the global topology view.

Mitigation: Under the proposed framework, each RID is bound to an explicit set of authorised (Net , $AreaID$, $Type$) tuples. When the verification module receives an LSA, it queries the smart contract with the extracted attributes. If the advertised prefix or any associated attribute does not match the registered policy for the corresponding RID , the smart contract returns *False*. The verification module then instructs the local router to reject the LSA, preventing it from influencing the LSDB or routing table.

Coverage and limitations: This mechanism detects any falsification that alters one or more of the four policy-controlled attributes. However, it does not detect attacks that manipulate dynamic fields (e.g., link metrics or sequence numbers) while keeping the policy-controlled attributes unchanged.

5.2. Adjacency-based injection attacks

Attack description: A compromised router exploits the trust established during OSPF neighbour formation to inject falsified LSAs after establishing a legitimate adjacency [12]. Standard OSPF authentication verifies neighbour identity but does not validate the semantic content of advertised routing information.

Mitigation: The proposed framework decouples routing authorisation from adjacency status. Every received LSA must pass policy verification regardless of whether the sender holds a valid adjacency. Consequently, even if an attacker establishes a legitimate OSPF adjacency and generates syntactically correct LSAs, any advertisement containing unauthorised prefixes, invalid flooding areas, or disallowed LSA types will fail policy validation and be rejected.

Coverage and limitations: The framework neutralises the attacker's ability to leverage adjacency trust for injecting unauthorised routing information. Attacks that exploit adjacency formation itself are outside the current scope.

5.3. Single-path injection attacks

Attack description: Single-path injection targets specific forwarding paths by crafting LSAs that appear to originate from a trusted router while advertising manipulated routing attributes [12]. The goal is to selectively redirect traffic through attacker-controlled nodes.

Mitigation: Policy enforcement binds each *RID* to a fixed set of authorised attributes. If an attacker forges an LSA using a legitimate *RID* but advertises a prefix or flooding area not associated with that *RID* in the policy table, the smart contract detects the mismatch and returns *False*. The forged advertisement is rejected before it can influence path computation.

Coverage and limitations: This mechanism prevents path manipulation through unauthorised attribute combinations. It does not address metric manipulation attacks where an attacker advertises authorised prefixes with artificially low costs to attract traffic, since link metrics are not currently included in the policy tuple.

5.4. Link-state database poisoning

Attack description. Structural poisoning corrupts the logical organisation of the LSDB by violating OSPF routing roles. Examples include a non-ABR router advertising inter-area Summary LSAs, a non-ASBR originating AS-External LSAs, or any router flooding LSAs into areas where it has no authorisation [10-12].

Mitigation: Because each policy entry explicitly associates an *RID* with permitted LSA types and

flooding areas, any role violation is detected during smart contract evaluation. A router attempting to originate Type 5 AS-External LSAs without ASBR authorisation, or flooding Summary LSAs into an area outside its permitted scope, will fail policy verification.

Coverage and limitations: The framework enforces structural constraints as defined by the administrator's policy. Its effectiveness depends on the completeness and correctness of the registered policies; missing policy entries could result in false negatives (unauthorised LSAs accepted) or false positives (legitimate LSAs rejected).

6. Experimental evaluation

6.1. Setup

The experimental topology (Fig. 2) comprises 15 OSPF routers across four areas: Area 0 (backbone, R5-R7), Area 1 (stub, R1-R3), Area 2 (normal, R10-R12), and Area 3 (NSSA, R13-R14). Three ABRs (R4, R8, R9) interconnect the areas, and one ASBR (R15) redistributes external routes into Area 3. The OSPF domain is emulated using EVE-NG with FRRouting. The smart contract is deployed on a private Ethereum network (Ganache, block gas limit 30,000,000). Verification modules communicate with the contract via Web3 JSON-RPC.

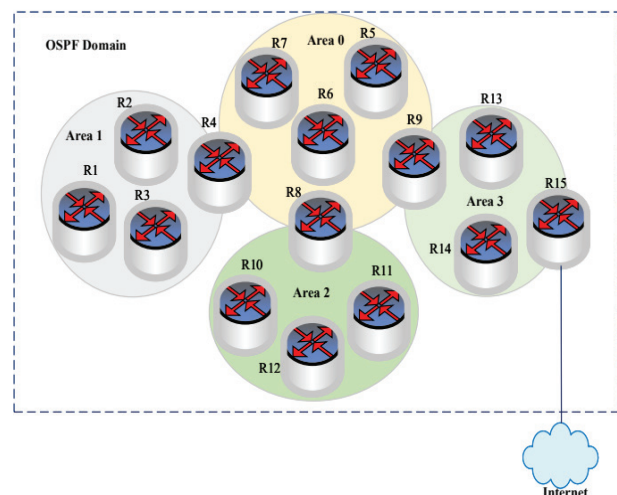


Fig. 2. Experimental OSPF network topology. Source: Authors' own work.

A total of 49 authorisation policies are registered on the smart contract, covering all legitimate (*Net*, *AreaID*, *Type*) tuples. The distribution comprises 39 Router LSAs (Type 1), 8 Summary LSAs (Type 3), and 2 AS-External LSAs (Type 5), as shown in Fig. 3.

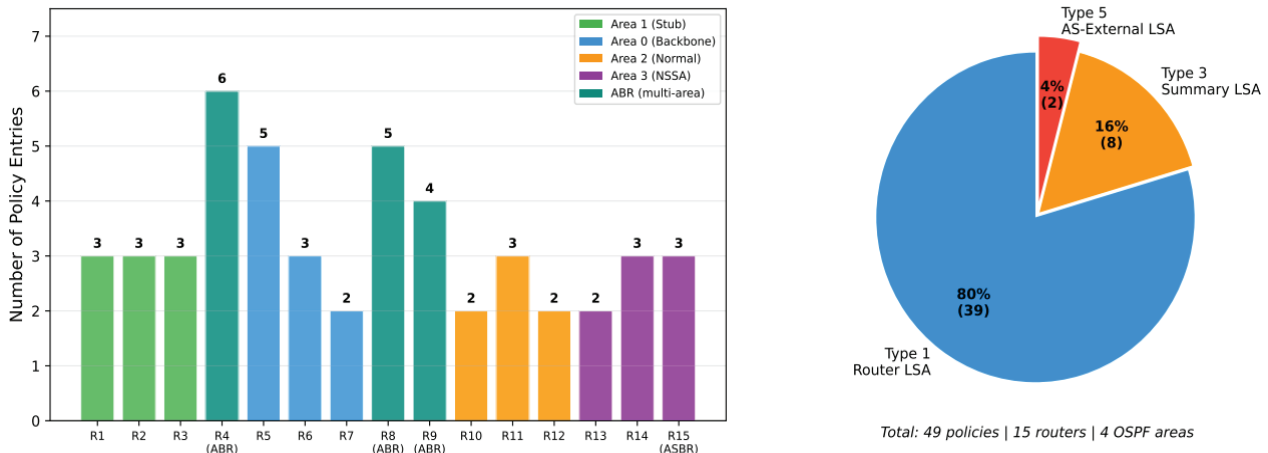


Fig. 3. Authorisation policy distribution across the experimental OSPF topology. (A) Number of policy entries per router; (B) Policy breakdown by LSA type. Source: Authors' own elaboration based on the experimental results.

6.2. Correctness

Eight scenarios evaluate each policy dimension and OSPF role, as summarised in Table 2.

Table 2. Evaluation scenarios and verification results.

#	Scenario	Violation	Result
S1	Authorised LSA (R1, Type 1, Area 1)	None	True
S2	Unauthorised prefix (R1, fake prefix)	Prefix	False
S3	Incorrect type (R10, Type 5 instead of 1)	Type	False
S4	Cross-area flood (R1, Area 0 instead of 1)	Area	False
S5	ABR spoofing (R6 sends Type 3)	Role	False
S6	ASBR spoofing (R13 sends Type 5)	Role	False
S7	ABR wrong-area summary (R8, Area 1)	Area	False
S8	Valid ABR summary (R4, Type 3, Area 0)	None	True

Source: Authors' own work.

All eight scenarios produced the expected result. Scenarios S5-S7 specifically validate role-based enforcement: non-ABR routers cannot generate Summary LSAs, non-ASBRs cannot originate AS-External LSAs, and ABRs are restricted to their authorised flooding areas.

6.3. Latency and gas cost

Latency analysis (Fig. 4A): The *PolVerification* function was invoked 100 times for each query type via off-chain *eth_call*. Fig. 4A reports three statistics for both authorised and unauthorised queries: the mean, the standard deviation (Std Dev), and the 95th percentile (P95). Authorised queries yielded a mean latency of 162.86 ms (Std Dev=28.70 ms, P95=216.42 ms). Unauthorised queries yielded a comparable mean of 158.09 ms (Std Dev=29.93 ms). The mean represents the average verification time per policy lookup. The standard deviation,

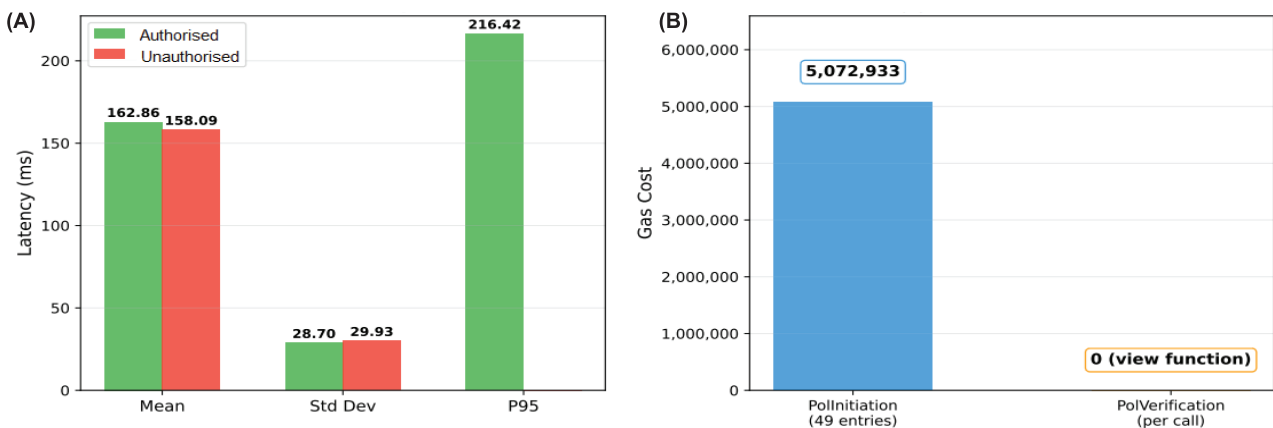


Fig. 4. Verification performance results. (A) Latency statistics for authorised and unauthorised queries; (B) On-chain gas cost comparison between PolInitiation and PolVerification. Source: Authors' own elaboration based on the experimental results.

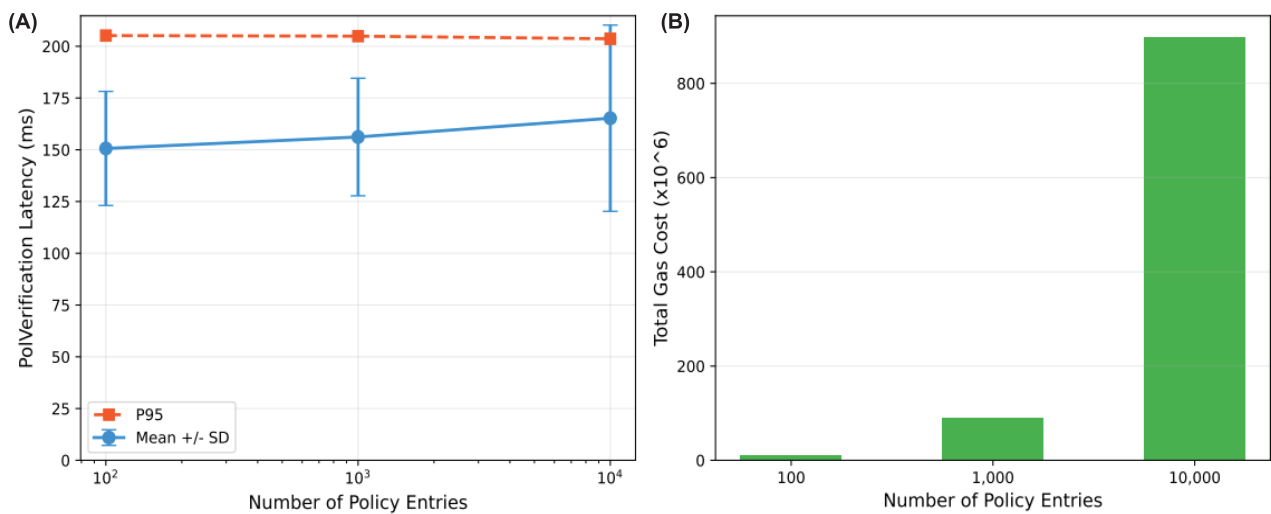


Fig. 5. Scalability evaluation results: (A) PolVerification latency as a function of policy table size; (B) Total PolInitiation gas cost as a function of policy count. Source: Authors' own elaboration based on the experimental results.

below 30 ms for both types, confirms low dispersion around the mean, indicating stable and predictable performance across invocations.

The P95 of 216.42 ms means that 95% of all authorised queries completed within this bound, providing a realistic worst-case estimate for deployment planning. The near-identical latency for both query types confirms constant-time execution regardless of the policy decision, attributable to the $O(1)$ hash-based lookup of Solidity's nested mapping structure. The P95 of 216.42 ms remains below OSPF's default SPF delay (200-5,000 ms) and far below the MinLSInterval of 5,000 ms [31], confirming that policy verification completes before SPF recalculation is triggered. The measured verification latency remains below typical OSPF timer thresholds and is therefore unlikely to dominate convergence behaviour in moderate-scale deployments. However, full convergence impact under large-scale or highly dynamic conditions requires further investigation.

Gas cost analysis: Fig. 4B compares the on-chain gas cost of the two primary operations. The *PolInitiation* function consumed a total of 5,072,933 gas for all 49 policy entries, averaging approximately 103,529 gas per entry. This is a one-time administrative cost incurred during initial policy deployment. The *PolVerification* function, implemented as a Solidity view function, reads blockchain state without modifying it. When invoked off-chain via *eth_call*, it incurs zero gas cost and generates no on-chain transaction. This design ensures that the most frequent operation, runtime

LSA verification, which is executed for every received advertisement, introduces no gas expenditure and no blockchain state modification. The on-chain cost is thus confined to infrequent administrative operations (policy initialisation and updates), while the runtime verification path remains cost-free.

7. Scalability evaluation

To evaluate the impact of policy table size on verification performance, synthetic policy entries were progressively loaded onto the smart contract at three scales: 100, 1,000, and 10,000 entries. At each scale, the *PolVerification* function was invoked 100 times for both authorised and unauthorised queries. Fig. 5 presents the results.

Verification latency (Fig. 5A): The mean verification latency remains near-constant across a 100-fold increase in policy table size: 150.64 ms at 100 entries, 156.17 ms at 1,000 entries, and 165.24 ms at 10,000 entries. The P95 latency is remarkably stable at approximately 204 ms across all three configurations (205.16, 204.88, and 203.61 ms, respectively), confirming that tail-latency behaviour is unaffected by policy table growth. The slightly elevated standard deviation at 10,000 entries (44.98 ms vs. 28 ms at smaller scales) is attributable to a single outlier measurement rather than systematic degradation, as the P95 remains unchanged.

Initialisation gas cost (Fig. 5B): The gas cost of *PolInitiation* scales linearly: 9.94 million gas for 100 entries, 89.83 million for 1,000, and 898.50 million for 10,000. The per-entry cost stabilises at approximately 89,850 gas under bulk initialisation,

with the higher per-entry cost at 100 entries (99,415 gas) reflecting proportionally greater fixed transaction overhead in smaller batches. The *PolVerification* function remains a zero-gas view function at all scales, ensuring that runtime cost is independent of policy table size. The initialisation cost, while growing linearly, is a one-time administrative expense that does not recur during normal OSPF operation.

8. Discussion

The proposed framework enforces routing authorisation by validating the static tuple (*RID*, *Net*, *AreaID*, *Type*) against blockchain-stored policies, enabling deterministic verification aligned with administrative intent.

Comparison with alternatives: A centralised policy server could offer similar functionality with lower latency but introduces a single point of trust and failure under insider threats. In contrast, the blockchain-based design distributes policy storage, preserves integrity under compromise, and provides an immutable audit trail, at the cost of increased complexity and infrastructure dependence. The blockchain-based design therefore represents a trade-off between decentralised policy consistency and additional infrastructure overhead, prioritising auditability and resilience against single-point compromise rather than minimal operational complexity.

Scope of protection: The framework focuses on static attributes and does not analyse dynamic LSA fields such as sequence numbers, age, or metrics. Attacks exploiting these parameters, e.g., sequence inflation or metric manipulation, remain out of scope and require complementary stateful monitoring.

Policy management: Effectiveness depends on policy completeness and correctness. Missing entries may allow unauthorised LSAs, while overly restrictive rules may block legitimate ones. Automated policy generation from router configurations is a necessary but unaddressed requirement for large-scale deployment.

Deployment considerations: Practical challenges include deploying verification modules per router, handling blockchain delays or partitions, and defining fail-open vs. fail-closed behaviour. Policy propagation latency across nodes also affects responsiveness. These trade-offs require validation in realistic environments.

9. Conclusions

This paper presented a zero trust framework for OSPF that enables policy-driven verification of link-state advertisements without modifying the protocol. Each LSA's static attributes (*RID*, *Net*, *AreaID*, *Type*) are validated against blockchain-based policies enforced via smart contracts. Security analysis shows mitigation of key insider attacks, including LSA falsification, adjacency-based injection, single-path injection, and LSDB poisoning. Experiments on a 15-router, four-area topology confirm correct policy enforcement across all scenarios, including role-based violations. Performance evaluation reports a mean latency of 162.86 ms (P95=216.42 ms), within OSPF timing constraints. Verification incurs zero runtime gas cost, while scalability tests up to 10,000 entries demonstrate near-constant latency, consistent with $O(1)$ lookup complexity. Initialisation cost scales linearly but is a one-time operation. Future work will address stateful monitoring and automated policy generation.

CRedit author statement

Chinh Huynh Nguyen: Conceptualisation, Methodology, Formal analysis, Writing - review and editing, Supervision; Tam Thanh Huynh, Doan Cong Dinh: Methodology, Software, Data curation, Formal analysis, Original draft preparation, Visualisation.

ACKNOWLEDGEMENTS

This research is funded by Ho Chi Minh City University of Technology and Engineering (HCMUTE) under grant number T2026-13.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] M. Goyal, M.S.M. Zahid, E. Baccelli, et al. (2012), "Improving convergence speed and scalability in OSPF: A survey", *IEEE Communications Surveys & Tutorials*, **14**(2), pp.443-463, DOI: 10.1109/SURV.2011.011411.00065.
- [2] M. Waqas, S.U.R. Malik, S. Akbar, et al. (2019), "Convergence time analysis of OSPF routing protocol using social network metrics", *Future Generation Computer Systems*, **94**, pp.62-71, DOI: 10.1016/j.future.2018.11.003.
- [3] M. Bhatia, V. Manral, M. Fanto, et al. (2009), "OSPFv2 HMAC-SHA cryptographic authentication", *RFC 5709*, Internet Engineering Task Force, 14pp, DOI: 10.17487/RFC5709.

- [4] M. Bhatia, S. Hartman, D. Zhang, et al. (2015), "Security extension for OSPFv2 when using manual key management", *RFC 7474*, Internet Engineering Task Force, 20pp, DOI: 10.17487/RFC7474.
- [5] S. Hartman, D. Zhang (2013), "Analysis of OSPF security according to the keying and authentication for routing protocols (KARP) design guide", *RFC 6863*, 18pp, Internet Engineering Task Force, DOI: 10.17487/RFC6863.
- [6] M. Conti, N. Dragoni, V. Lesyk (2016), "A survey of man-in-the-middle attacks", *IEEE Communications Surveys & Tutorials*, **18**(3), pp.2027-2051, DOI: 10.1109/COMST.2016.2548426.
- [7] M. Gupta, N. Melam (2006), "Authentication/confidentiality for OSPFv3", *RFC 4552*, Internet Engineering Task Force, 15pp, DOI: 10.17487/RFC4552.
- [8] R. Coltun, D. Ferguson, J. Moy, et al. (2008), "OSPF for IPv6", *RFC 5340*, Internet Engineering Task Force, 94pp, DOI: 10.17487/RFC5340.
- [9] G. Nakibly, A. Kirshon, D. Gonikman, et al. (2012), "Persistent OSPF attacks", *Proc. Network and Distributed System Security Symposium (NDSS)*, 21pp.
- [10] G. Nakibly, A. Sosnovich, E. Menahem, et al. (2014), "OSPF vulnerability to persistent poisoning attacks: A systematic analysis", *Proc. 30th Annual Computer Security Applications Conference (ACSAC)*, pp.336-345, DOI: 10.1145/2664243.2664278.
- [11] E. Kaffashi, A.M. Mousavi, H.R. Rahvard, et al. (2014), "A new attack on link-state database in open shortest path first routing protocol", *Journal of Electrical and Electronic Engineering*, **3**(2-1), pp.39-45, DOI: 10.11648/j.jeees.2015030201.19.
- [12] Y. Song, S. Gao, A. Hu, et al. (2017), "Novel attacks in OSPF networks to poison routing table", *Proc. IEEE International Conference on Communications (ICC)*, pp.1-6, DOI: 10.1109/ICC.2017.7996829.
- [13] B. Al-Musawi, P. Branch, M.F. Hassan (2020), "Identifying OSPF LSA falsification attacks through non-linear analysis", *Computer Networks*, **167**, DOI: 10.1016/j.comnet.2019.107031.
- [14] N. Devir, O. Grumberg, S. Markovitch, et al. (2019), "Topology-agnostic runtime detection of OSPF routing attacks", *Proc. IEEE Conference on Communications and Network Security (CNS)*, pp.277-285, DOI: 10.1109/CNS.2019.8802826.
- [15] B. Al-Musawi, P. Branch (2018), "Identifying OSPF anomalies using recurrence quantification analysis", *arXiv*, DOI: 10.48550/arXiv.1805.08087.
- [16] S. Jin, Z. Zhang, K. Chakrabarty, et al. (2019), "Anomaly detection and health-status analysis in a core router system", *IEEE Design and Test*, **36**, pp.7-17, DOI: 10.1109/MDAT.2019.2906108.
- [17] S. Sengupta, H. Kim, J. Rexford (2022), "Continuous in-network round-trip time monitoring", *Proc. ACM SIGCOMM 2022 Conference*, pp.473-485, DOI: 10.1145/3544216.3544222.
- [18] P. Kasemsuwan, V. Visoottiviseth (2017), "OSV: OSPF vulnerability checking tool", *Proc. 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp.1-6, DOI: 10.1109/JCSSE.2017.8025919.
- [19] K. Obelovska, Y. Snaichuk, J. Selecky, et al. (2023), "An approach toward packet routing in the OSPF-based network with a distrustful router", *WSEAS Transactions on Information Science and Applications*, **20**, pp.432-443, DOI: 10.37394/23209.2023.20.45.
- [20] K. Obelovska, O. Tkachuk, Y. Snaichuk (2024), "Minimizing the number of distrustful nodes on the path of IP packet transmission", *Computation*, **12**(5), DOI: 10.3390/computation12050091.
- [21] K. Obelovska, Y. Snaichuk, O. Liskevych, et al. (2025), "Mitigation of risks associated with distrustful routers in OSPF networks: An enhanced method", *Computers*, **14**(2), DOI: 10.3390/computers14020043.
- [22] S. Othmen, W. Mansouri, S. Askany (2024), "Robust and secure routing protocol based on group key management for Internet of Things systems", *Engineering, Technology & Applied Science Research*, **14**(3), pp.14402-14410, DOI: 10.48084/etasr.7115.
- [23] J. Pereira, T. Klenze, S. Giampietro, et al. (2025), "Protocols to code: Formal verification of a secure next-generation Internet router", *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp.1469-1483, DOI: 10.1145/3719027.3765104.
- [24] G.V. Lakshmi, P. Vaishnavi (2022), "An efficient security framework for trusted and secure routing in MANET: A comprehensive solution", *Wireless Personal Communications*, **124**(1), pp.333-348, DOI: 10.1007/s11277-021-09359-2.
- [25] X. Shao, L. Gao (2020), "Verifying policy-based routing at internet scale", *Proc. IEEE INFOCOM 2020*, pp.2293-2302, DOI: 10.1109/INFOCOM41043.2020.9155235.
- [26] Y. Bai, X. Zhang, D. Yu, et al. (2022), "A deep reinforcement learning-based geographic packet routing optimization", *IEEE Access*, **10**, pp.107421-107431, DOI: 10.1109/ACCESS.2022.3213649.
- [27] A. Kumar, V.K. Mishra, U.S. Yadav, et al. (2025), "A secure framework for source routing in autonomous systems", *Proc. International Conference on Recent Advancements in Artificial Intelligence*, pp.67-77, DOI: 10.1007/978-981-96-7760-3_5.
- [28] S. Rose, O. Borchert, S. Mitchell, et al. (2020), "Zero trust architecture", *National Institute of Standards and Technology, NIST Special Publication 800-207*, 59pp, DOI: 10.6028/NIST.SP.800-207.
- [29] Z. Zheng, S. Xie, H.N. Dai, et al. (2018), "Blockchain challenges and opportunities: A survey", *International Journal of Web and Grid Services*, **14**(4), pp.352-375, DOI: 10.1504/IJWGS.2018.095647.
- [30] M. Conti, E.S. Kumar, C. Lal, et al. (2018), "A survey on security and privacy issues of Bitcoin", *IEEE Communications Surveys & Tutorials*, **20**(4), pp.3416-3452, DOI: 10.1109/COMST.2018.2842460.
- [31] J. Moy (1998), "OSPF version 2", *RFC 2328*, Internet Engineering Task Force, DOI: 10.17487/RFC2328.

Application of convolutional neural networks to enhance classification accuracy of synthetic aperture radar ground targets

Huu Cuong Thieu¹, Hieu Nguyen^{2*}, Ngoc Binh Nguyen¹

¹Telecommunications University, 101 Mai Xuan Thuong Street, Bac Nha Trang Ward, Khanh Hoa Province, Vietnam

²Air Force's Officer College, Biet Thu Street, Nha Trang Ward, Khanh Hoa Province, Vietnam

Received 6 March 2026; revised 24 June 2026; accepted 18 August 2026

Abstract:

The paper proposes a convolutional neural network (CNN) approach for identifying ground target features from acquired synthetic aperture radar (SAR) imagery. Experimental results demonstrate that the proposed CNN model outperforms traditional machine learning methods in terms of classification accuracy and generalisation capability. Specifically, the CNN achieves an accuracy of 99.65%, compared with 94.59% for support vector machine (SVM), 92.98% for K-nearest neighbors (K-NN), and 90.12% for decision tree (DT). On the other hand, the proposed CNN model obtains an average accuracy of 99.65% on the testing dataset, which is highly consistent with the validation accuracy of 99.77%. The small difference between the validation and testing results indicates that the model maintains stable performance when applied to unseen data. These findings demonstrate the effectiveness of CNN-based feature identification for SAR imagery and highlight its strong generalisation capability. The proposed approach therefore provides a reliable and robust solution for automated ground target recognition in SAR image analysis and related remote sensing applications.

Keywords: classification accuracy, convolutional neural network (CNN), synthetic aperture radar (SAR).

1. Introduction

Currently, various methods are employed for airborne radar surface target detection. Among these, SAR has become the most widely used and effective approach. The spatial and temporal heterogeneity, along with the topographical characteristics of SAR imagery, serve as critical features for target classification. Driven by rapid technological advancements, the number and diversity of ground military vehicles have increased significantly, posing a major challenge for precise target identification. Several machine learning methods have been previously proposed, such as the support vector machine (SVM) method [1, 2], K-NN algorithm [2, 3], and DT [4]. Although these methods yield certain results, they still possess several limitations. Developing classification models using these algorithms often faces challenges when dealing with large-scale datasets, multi-label classification, or multi-domain data. Another drawback is their heavy dependence on the dataset; even minor changes in the data can lead to complete shifts in the model structure. Furthermore, these methods require accessing the entire training set to calculate

the differences between them and the samples being classified, resulting in a high computational workload and significant consumption of system memory resources.

Specifically designed to handle image data, pattern recognition, and spatial features, CNNs generally achieve higher accuracy than K-NN, SVM, or DT. Furthermore, they exhibit superior generalisation across large-scale datasets due to their ability to learn abstract representations through multiple hierarchical layers [5, 6]. This paper proposes a target recognition model based on a CNN architecture, where raw imagery from airborne radar is fed directly into the network for training. Experimental results demonstrate that the proposed model achieves high accuracy and significant reliability in its predictive performance.

The remainder of this paper is organised as follows: Section 2 describes the proposed CNN model architecture. The simulation results and performance evaluations are presented in Section 3. Finally, Section 4 provides the conclusion and suggests future research directions.

*Corresponding author: Email: nguyenhieu25mta@gmail.com

2. Proposed methodology

2.1. Convolutional neural network

CNN represent one of the most advanced deep learning architectures, widely utilised in the fields of image and speech recognition and classification [6-8]. These networks are designed to learn salient features from input data through convolutional layers. Since pixel information typically influences only its immediate neighbors, convolutional filters allow the network to focus on small, localised regions of an image during each computational step. This mechanism significantly reduces the number of parameters required for optimisation during the training process. Consequently, CNN are capable of efficiently extracting local features from input data, thereby reducing model complexity and enhancing both computational speed and overall model performance.

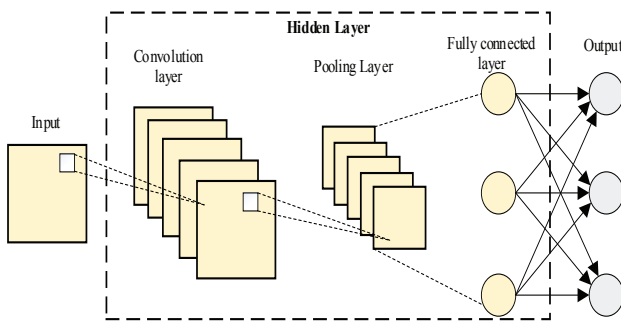


Fig. 1. Network structure.

Figure 1 illustrates a typical CNN architecture in practice. The convolutional layer performs convolution operations on the input data using a set of filters to extract features. Each filter consists of a weight matrix corresponding to a specific feature of the input. A filter is a matrix that slides across each region of the input matrix, performing element-wise multiplication and summing the resulting values. The positions of the elements in the output matrix must correspond to the sliding sequence of the filter. Fig. 2 provides an example of the sequence and the method of calculating these convolutions.

The pooling layer is used after the convolutional layer with the primary goal of reducing the spatial dimensions (width and height) of the input data while preserving its essential features. Generally, two pooling methods are employed: max pooling and average pooling. Fig. 3 illustrates examples of these two downsampling techniques.

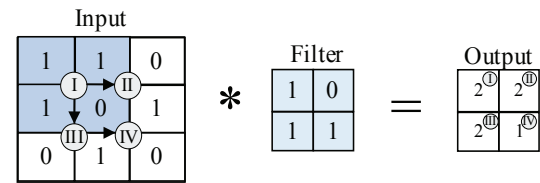


Fig. 2. Example of Calculating the image feature matrix.

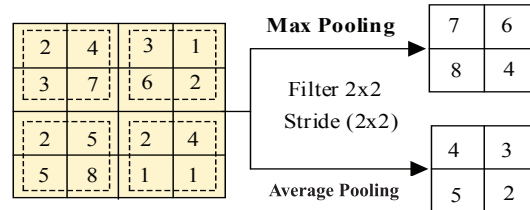


Fig. 3. Example of two pooling methods.

The fully connected layer is typically situated at the final stage of a neural network. Its primary function is to perform classification and prediction based on the features extracted by the preceding convolutional and pooling layers. In a fully connected layer, every input is linked to every neuron; consequently, each output in this layer depends on the entire set of inputs from the previous sub-layer.

2.2. Proposed convolutional neural network model

Determining the optimal architecture for a neural network depends heavily on its specific application domain. In this paper, we propose an advanced neural network architecture incorporating convolutional layers to classify eight types of ground targets. The structure and specific parameters of the proposed CNN model are detailed in Fig. 4 and Table 1.

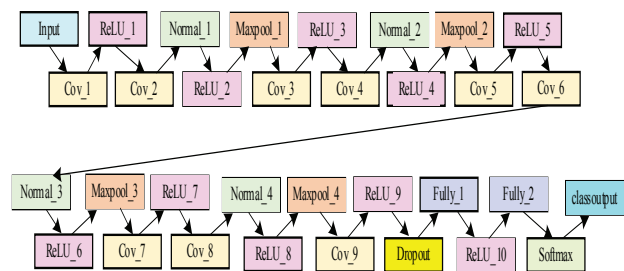


Fig. 4. Proposed convolutional neural network model structure.

The input signals consist of a radar acquired image dataset, indexed with a fixed dimension of 128x128x1.

The signal is passed through convolutional blocks using a 1-channel 2D convolution operation. The downsampling layer is constructed using the max pooling method. An activation function is an

indispensable component in every convolutional block. Real-world data has a highly complex distribution; if only convolutional operations (linear representation) were used, it would not accurately reflect the actual model. Therefore, activation functions are utilised to create non-linear outputs from the input data. Some commonly used activation functions include Sigmoid, Tanh, ReLU, Leaky ReLU, Maxout, etc. In the proposed model, the ReLU function is used to set negative input values to zero while keeping the input values unchanged when they are greater than zero.

Table 1. Proposed CNN model parameters.

Layer	Filter	Output
Input	-	128x128x1
Cov_1	32@(3x3), padding 'same'	128x128x32
Relu_1	-	128x128x32
Cov_2	32@(3x3) padding 'same'	128x128x32
Norm_2, Relu_2	-	128x128x32
Maxpool_2	2x2	64x64x32
Cov_3	64@(3x3) padding, 'same'	64x64x64
Relu_3	-	64x64x64
Cov_4	64@(3x3) padding 'same'	64x64x64
Norm_4, Relu_4	-	64x64x64
Maxpool_4	2x2	32x32x64
Cov_5	128@(3x3) padding 'same'	32x32x128
Relu_5	-	32x32x128
Cov_6	128@(3x3) padding 'same'	32x32x128
Norm_6, Relu_6	-	32x32x128
Maxpool_6	2x2	16x16x128
Cov_7	256@(3x3) padding 'same'	16x16x256
Relu_7	-	16x16x256
Cov_8	256@(3x3) padding 'same'	16x16x256
Norm_8, Relu_8	-	16x16x256
Maxpool_8	2x2	8x8x256
Cov_9	512@(6x6) Padding 'valid'	3x3x512
Relu_9	-	3x3x512
Dropout	-	3x3x512
Fully_1(512)	-	512x1x1
Relu_10	-	512x1
Fully_2(8)	-	8x1
Softmax, classoutput	-	8x1

Next is the batch normalisation layer, which is responsible for normalizing input values to accelerate the convergence of the deep learning network in a stable manner. The normalisation process is performed by subtracting the mean (μ) and dividing by the standard deviation (σ) according to the formula 1.

$$\text{Norm}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (1)$$

The number of hidden layers in the network helps improve accuracy; however, the number of operations becomes more complex and training time increases. Therefore, to balance these requirements, the proposed model utilises 10 hidden layers.

The output size is maintained equal to the input size by using a stride of 1 and 'same' padding. In the proposed model, the max pooling method is used for pooling with 2x2 window size and a stride of (2x2). A regularisation technique is applied during network training using a Dropout(0.5) function. The output data at the end of the model is flattened into a vector. In MATLAB, for 2D image input data, the fullyConnectedLayer automatically flattens all outputs from previous layers into a vector. The fully_2 layer is configured with 8 outputs, corresponding to the 8 targets to be classified. This 8-element vector then passes through a Softmax layer to calculate the probability of occurrence for each of the 8 predefined target types. This probability is used to predict the final output.

It is expected that the proposed model will demonstrate strong generalisation capabilities and achieve higher accuracy compared to traditional models. Parameters such as the number of layers, filter sizes, and network depth have been empirically tuned to optimise the network's performance.

3. Simulation results and discussion

In this section, experiments are conducted on a real-world dataset obtained from the MSTAR data source, implemented using MATLAB 2022b software within a GPU 630 execution environment. The dataset consists of 8,688 SAR images from 7 ground vehicles (BMP2 infantry fighting vehicle, BTR70 armoured personnel carrier, T62 tank, ZSU-23/4 self-propelled anti-aircraft gun, 2S1 self-propelled howitzer, ZIL-131 truck, and D7 bulldozer) and 1

military structure. The images were captured using X-band SAR at various depression angles (15, 16, 17, 29, 30, 31, 43, 44 and 45°) with approximately 190 to 300 versions at different azimuth angles, providing comprehensive coverage over 360° (Fig. 5).

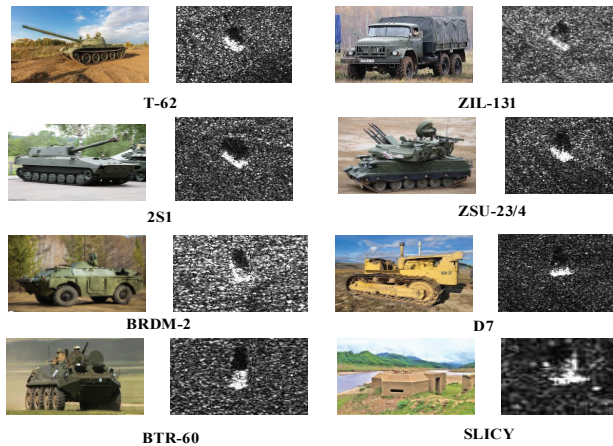


Fig. 5. Optical and synthetic aperture radar images of targets. Source: Author's compilation.

Next, several parameters for the training process are established. The input data matrix is defined as [128x128x1]. The mini-batch size used for weight updates is set to 48. A larger mini-batch size accelerates the computation process because operations are performed in parallel; however, it requires more memory resources. Experiments demonstrate that with the same number of epochs, a mini-batch size of 48 is optimal in terms of both accuracy and training time.

To monitor the accuracy during training, the validation data is specified with a validation frequency of 15. The network training process is performed on the training dataset, while the accuracy is calculated on the validation data at regular intervals of every 15 iterations throughout the training process, with data shuffling performed after each epoch. The validation data is not used to update the network weights. The initial learning rate is set to 0.001, and the stochastic gradient descent with momentum (SGDM) optimiser is employed.

The number of training cycles (epochs) must be selected appropriately as it directly impacts the model's accuracy. The model requires a sufficiently large number of epochs to learn the general structure of the data. However, when the number of epochs increases but the accuracy plateaus, it indicates that the model has reached its optimum and further training is unnecessary. With the number of training epochs set to n , the total number of iterations (the number of weight updates) during the training process is determined by:

$$\text{Iterations} = \frac{8688 \times 0.8}{48} \times n \quad (2)$$

The experimental results in Fig. 6 show that from the 8th training epoch onward, the accuracy of the training model remains nearly unchanged (99.77% compared to 99.89%). This indicates that the model has reached its optimum and further training is unnecessary. Therefore, to balance accuracy and training time, 8 epochs are recommended for this model.

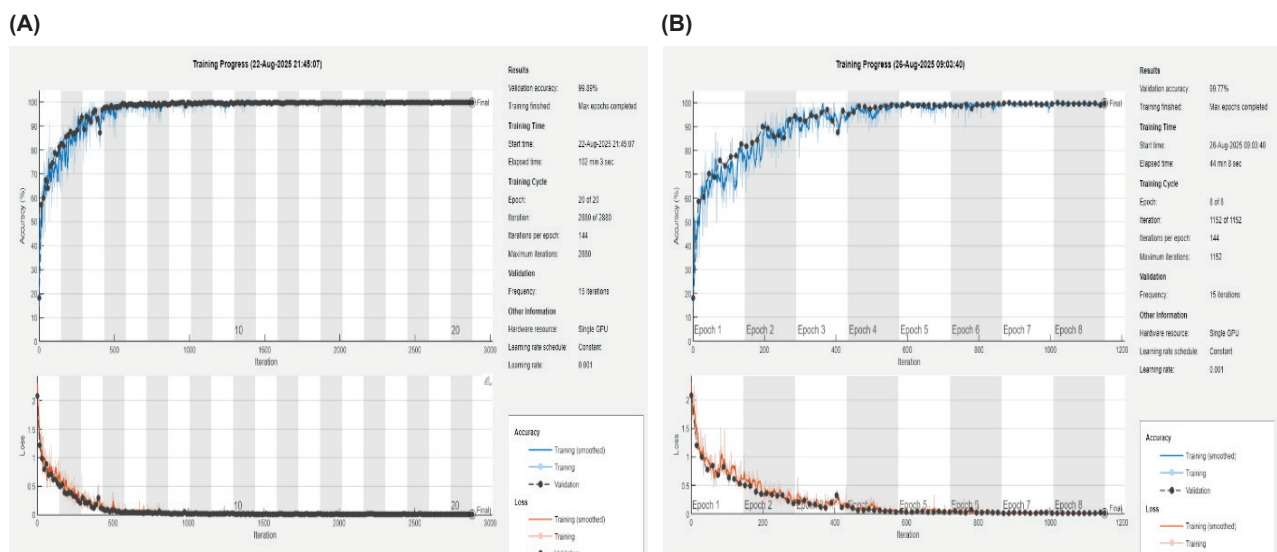


Fig. 6. Training accuracy and loss for 20 epochs (A) and 8 epochs (B).

Testing accuracy indicates the predictive capability of the trained model and is determined by the ratio of correctly classified samples to the total number.

$$A_{cc} = \frac{N_{ex}}{N} \times 100\% \quad (3)$$

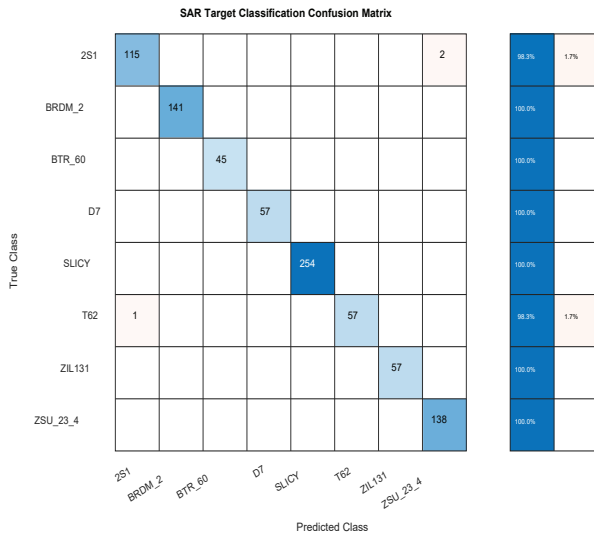
In this context, A_{cc} represents the accuracy (%), N_{ex} is the number of correctly predicted samples, and N is the total number of samples in the testing set.

To further evaluate the performance of the CNN, traditional classifiers such as SVM, KNN, and

DT were designed for comparison. Experimental results show that the SVM method achieves the highest accuracy when using an OvO (One-vs-One) binary coding training set with a non-linear SVM (RBF kernel) and a penalty parameter $C=100$. The K-NN method is configured as a predictor with 11 nearest neighbours and the Minkowski distance metric. Finally, the DT method is implemented with a maximum of 200 splits.

To provide a detailed evaluation of the classification accuracy for each specific target between the proposed model and the traditional models, a confusion matrix is utilised in this paper.

(A)



(B)

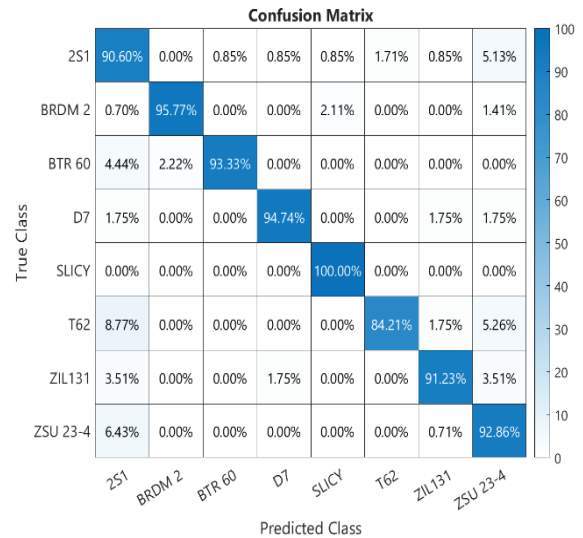
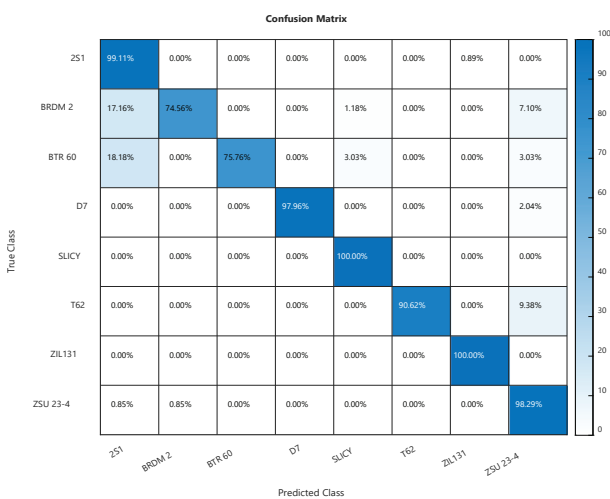


Fig. 7. Confusion matrices for target classification of the proposed CNN model (A) and the SVM model (B).

(A)



(B)

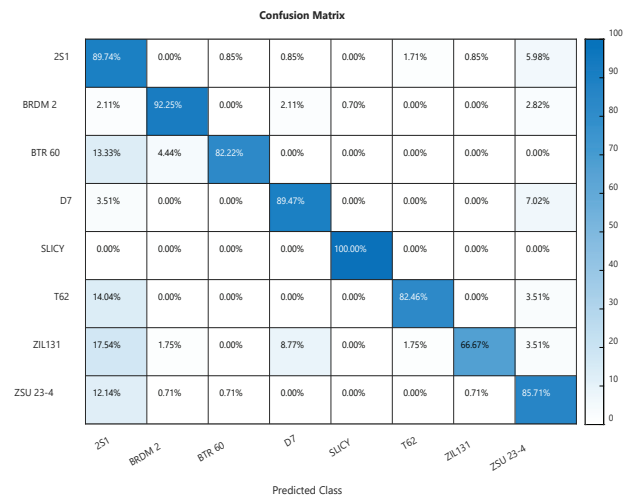


Fig. 8. Confusion matrices for target classification of the KNN (A) and DT model (B).

Table 2. Accuracy of the proposed CNN model compared to traditional models.

No.	Method	Accuracy
1	CNN	99.65%
2	SVM	94.59%
3	K-NN	92.98%
4	DT	90.12%

The results presented in Table 2, Figs. 7 and 8 demonstrate that the proposed model significantly improves the classification accuracy for SAR image targets. Furthermore, the proposed CNN model achieved an average accuracy of 99.65% on the testing dataset, which is remarkably close to the validation accuracy of 99.77%. This consistency ensures the reliability and robustness of the model.

Currently, several advanced deep learning models, such as the Multi-Stream Convolutional Neural Network (MS-CNN), also achieve high accuracy in SAR image classification (99.84% for 2-view MS-CNN, 99.88% for 3-view MS-CNN, and 99.92% for 4-view MS-CNN). However, their computational complexity in terms of (FLOPs) is very high (5.044×10^7 ; 7.566×10^7 ; and 1.008×10^8) [9]. This highlights the superiority of the proposed model, as it maintains high accuracy while significantly reducing computational complexity.

4. Conclusions

This paper proposes an enhanced method for SAR image target classification by employing a CNN as an alternative to traditional models. The parameters of the proposed architecture are detailed specifically through each convolutional layer. Experimental results indicate that the proposed method significantly improves classification accuracy and model reliability, achieving 99.65%, compared to 94.59% for SVM, 92.98% for K-NN, and 90.12% for DT. Further theoretical and experimental research will be conducted in the future by varying the signal-to-noise ratios (SNRs) and integrating target-specific variations, such as environmental conditions, into the training dataset. Additionally, we will extend the accuracy evaluation of the proposed model to larger datasets, such as MSAR-1.0 and SAR-AIRCRAFT-1.0, to enhance the model's generalisation capability and further improve its performance. Furthermore, we plan to incorporate adaptive hyperparameter optimisation methods (such as Bayesian optimisation or Neural Architecture Search - NAS)

in our future work to systematically evaluate and enhance the model's robustness across varying SAR sensor resolutions.

CRediT author statement

Huu Cuong Thieu: Conceptualisation, Methodology, Mathematical analysis, Writing - Reviewing, and Supervision; Hieu Nguyen: Formal analysis, Mathematical modelling, Validation, Writing (original draft), Simulation implementation, Result interpretation, Writing - Review and Editing; Ngoc Binh Nguyen: Simulation design, Performance evaluation, Data curation.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] S. Kent, N.G. Kasapoglu, M. Kartal (2008), "Radar target classification based on support vector machines and high resolution range profiles", *IEEE Radar Conference*, DOI: 10.1109/RADAR.2008.4721107.
- [2] M.E. Demirhan, Ö. Salor (2016), "Classification of targets in SAR images using SVM and K-NN techniques", *IEEE 24th Signal Processing and Communication Application Conference (SIU)*, pp.1581-1584, DOI: 10.1109/SIU.2016.7496056.
- [3] D.S. Cao, Y.Z. Liang, L.X. Zhang, et al. (2012), "Kernel k -nearest neighbor algorithm as a flexible SAR modeling tool", *Chemometrics and Intelligent Laboratory Systems*, **144**, pp.19-23, DOI: 10.1016/j.chemolab.2012.01.008.
- [4] S. Lee, C. Lee, K.G. Mun, et al. (2022), "Decision tree algorithm considering distances between classes", *IEEE Access*, **10**, pp. 69750-69756, DOI: 10.1109/ACCESS.2022.3187172.
- [5] M. Yani, F. Marpaung, M. Hidayat, et al. (2022), "A comparative study of convolutional neural network and k -nearest neighbours algorithms for food image recognition", *Computational Technologies*, **27(6)**, pp.88-99, DOI: 10.25743/ICT.2022.27.6.008.
- [6] L. Miaoge, C. Bo, W. Dongsheng, et al. (2024), "CNN model visualisation method for SAR image target classification", *Journal of Radars*, **13(2)**, pp.359-373, DOI: 10.12000/JR23107.
- [7] Z. Huang, X. Yao, Y. Liu, et al. (2022), "Physically explainable CNN for SAR image classification", *ISPRS Journal of Photogrammetry and Remote Sensing*, **190**, pp.25-37, DOI: 10.1016/j.isprsjprs.2022.05.008.
- [8] J. Lu, S. Song, Z. Hu (2020), "Fundamental frequency detection of underwater acoustic target using demon spectrum and cnn network", *3rd International Conference on Unmanned Systems*, pp.778-784, DOI: 10.1109/ICUS50048.2020.9274949.
- [9] P. Zhao, K. Liu, H. Zou, et al. (2018), "Multi-stream convolutional neural network for sar automatic target recognition", *MDPI Journal Remote Sensing*, **10(9)**, DOI: 10.3390/rs10091473.

Vietnamese interior image captioning: A new dataset and fine-tuned BLIP-2 model for vision-language generation

Cong Minh Vu, Quang Hung Nguyen*

Post and Telecommunications Institute of Technology, Km10, Nguyen Trai Street, Ha Dong Ward, Hanoi, Vietnam

Received 8 January 2026; revised 15 April 2026; accepted 10 June 2026

Abstract:

This article investigates automatic image captioning for furniture products in Vietnamese, a low-resource language with complex tokenisation characteristics. While recent vision-language models perform strongly on English benchmarks, their applicability to Vietnamese and domain-specific scenarios remains underexplored. To address this gap, we construct a domain-specific dataset of 3,483 furniture product images paired with Vietnamese descriptions emphasising key attributes such as product type, material, size, and colour, publicly available to support reproducibility and future research. Based on this dataset, we study the adaptation of BLIP-2, a pretrained vision-language model, under limited data conditions. Our fine-tuning strategy preserves the original tokeniser, employs BF16 precision for stable optimisation, and carefully adjusts generation hyperparameters to balance descriptive richness and factual accuracy. Through extensive experiments, we analyse the model's behaviour in terms of tokenisation, generation length, and hallucination. Under the optimal configuration (maximum length=70, length penalty=1.5), our approach achieves BLEU-4 of 31.22, CIDEr of 243.60, METEOR of 35.49, and ROUGE-L of 60.47 on the test set, substantially outperforming zero-shot baselines including LLaVA-1.5-7B (7.3B parameters) and InstructBLIP. The findings offer practical insights into applying vision-language models to low-resource languages and specialised domains, laying a foundation for future research and real-world deployment in Vietnamese e-commerce.

Keywords: BLIP-2, domain-specific dataset, furniture domain, image captioning, Vietnamese, vision-language models.

1. Introduction

Image captioning, which aims to generate natural language descriptions from visual inputs, plays a crucial role at the intersection of computer vision and natural language processing. This task has attracted significant attention due to its wide range of applications, including assistive technologies, image retrieval, and automated content creation for e-commerce platforms. Recent advances in vision-language models have substantially improved captioning performance on standard benchmarks, largely driven by large-scale pretraining on English-centred datasets [1-3].

Despite these advances, extending image captioning systems to low-resource languages

remains a challenging problem. Vietnamese, in particular, presents unique linguistic characteristics, such as complex diacritics, flexible word segmentation, and rich compound expressions, which are not well aligned with tokenisers and language models predominantly trained on English data [4, 5]. As a result, directly applying pretrained vision-language models to Vietnamese often leads to unstable training or degraded generation quality.

In addition to language-related challenges, domain specificity further complicates the image captioning task. In the context of furniture e-commerce, product descriptions are expected to convey precise and factual information, including product category, material, dimensions, and colour. Unlike generic image captions, furniture

*Corresponding author: Email: hungnq1@ptit.edu.vn

descriptions require higher accuracy and greater attention to fine-grained attributes, making domain adaptation a critical requirement. However, existing Vietnamese image captioning datasets primarily focus on general-purpose images and are not suitable for such specialised applications.

Motivated by these challenges, this study aims to explore the following research questions:

Q1: How effectively can a pretrained vision-language model adapt to Vietnamese image captioning in a domain-specific furniture setting with limited training data?

Q2: What is the impact of Vietnamese tokenisation strategies on the performance and stability of vision-language models?

Q3: How do generation-related design choices influence the trade-off between descriptive completeness and hallucination in domain-specific captioning?

To address these questions, we construct a Vietnamese furniture image captioning dataset and investigate the adaptation of the BLIP-2 model using a tailored fine-tuning strategy. Rather than proposing a new architecture, our work focuses on understanding how existing state-of-the-art models behave when applied to a low-resource language and a specialised domain.

The main contributions of this paper are summarised as follows:

- We introduce a domain-specific Vietnamese dataset for furniture image captioning, consisting of 3,483 high-quality image-caption pairs. The dataset is publicly available at Kaggle to facilitate reproducibility.

- We propose a practical fine-tuning strategy for adapting BLIP-2 to Vietnamese without modifying the original tokeniser, suitable for limited-data scenarios.

- We provide a comprehensive experimental evaluation including five zero-shot baseline configurations across four model families (GIT with two Vietnamese adaptation strategies, BLIP-2, InstructBLIP, and LLaVA-1.5-7B), quantitative tokenisation analysis, systematic error classification, and a hyperparameter sensitivity analysis.

- We offer empirical insights that can guide future research and real-world applications of vision-language models for low-resource languages.

2. Related work

This section reviews prior studies relevant to our work, focusing on three main research directions: image captioning models, vision-language pretrained models, and image captioning for low-resource languages and domain-specific applications. We then summarise the research gap addressed by this study.

2.1. Image captioning models

Early approaches to image captioning primarily relied on encoder-decoder architectures that combine convolutional neural networks (CNNs) for visual feature extraction with recurrent neural networks (RNNs), such as Long Short-Term Memory (LSTM), for sentence generation. Representative works include the Show and Tell model, which introduced an end-to-end framework for mapping images to textual descriptions [6], and subsequent attention-based models that improved performance by allowing the decoder to focus on salient image regions during generation [7, 8].

With the advent of the Transformer architecture, image captioning models shifted toward attention-only mechanisms [9], enabling more effective modelling of long-range dependencies and richer visual-textual interactions. Transformer-based captioning models demonstrated improved performance on standard benchmarks by replacing recurrent structures with multi-head self-attention and cross-attention mechanisms [10, 11]. Despite these improvements, most of these models are trained from scratch or with limited pretraining, making them heavily dependent on large annotated datasets and less suitable for low-resource settings.

2.2. Vision-language pretrained models

Recent progress in image captioning has been driven by large-scale vision-language pretraining. Models such as CLIP introduced contrastive learning on massive image-text pairs, enabling strong visual-textual alignment across diverse domains [2]. Building on this idea, generative vision-language models such as GIT and BLIP further unified understanding and generation tasks by jointly learning image-text representations.

BLIP-2 represents a significant advancement in this line of research by introducing a lightweight Querying Transformer (Q-Former) that bridges a frozen image encoder and a frozen large language model [12]. This design allows BLIP-2 to leverage powerful pretrained components while requiring

only a small fraction of parameters to be fine-tuned. As a result, BLIP-2 achieves strong performance across multiple vision-language tasks with reduced computational cost.

More recently, instruction-tuned vision-language models have emerged as a new paradigm. LLaVA connects a vision encoder to a large language model through a projection layer, enabling visual instruction following [13]. InstructBLIP extends BLIP-2 with instruction-aware fine-tuning for improved zero-shot generalisation [14]. Despite their impressive capabilities on English tasks, these models remain primarily optimised for high-resource languages.

In the domain adaptation context, recent work has explored parameter-efficient transfer learning for adapting vision-language models to specialised domains. For example, X. Zhao, et al. (2025) [15] adapted BLIP-2 to remote sensing image captioning, demonstrating the flexibility of the Q-Former architecture for cross-domain transfer.

However, most vision-language pretrained models are primarily trained on English-centred data. Their tokenisers and language modelling components are optimised for English or high-resource languages, which can lead to degraded performance when directly applied to languages with different linguistic structures. This limitation is particularly pronounced for generative tasks such as image captioning, where errors in tokenisation can propagate and severely affect output quality.

2.3. Image captioning for low-resource languages and domain-specific applications

Research on image captioning for low-resource languages remains relatively limited. Existing datasets and benchmarks, such as those for Vietnamese image captioning, are mostly constructed from general-domain images and contain relatively simple descriptions [16, 17]. While these datasets provide a useful starting point, they do not capture the complexity required in specialised domains such as e-commerce, where captions must be both descriptive and factually precise.

Parallel to this, domain-specific image captioning has been explored in fields such as medical imaging, food recognition, and fashion [18-20]. These studies highlight the importance of domain adaptation, as generic captioning models often fail to capture specialised terminology and fine-grained attributes. Common approaches include collecting domain-

specific datasets and fine-tuning pretrained models accordingly. Nevertheless, such methods typically assume the availability of sufficient annotated data and do not explicitly address challenges arising from language-specific tokenisation.

In the Vietnamese context, domain-specific image captioning for furniture products has not been systematically studied. The lack of publicly available datasets and benchmarks further limits progress in this area, creating a barrier to both academic research and practical deployment.

2.4. Summary and research gap

In summary, prior work has demonstrated the effectiveness of vision-language models for image captioning, particularly through large-scale pretraining and Transformer-based architectures. However, existing approaches largely focus on English or high-resource languages and general-purpose datasets. Research on adapting these models to low-resource languages, such as Vietnamese, and to domain-specific scenarios remains scarce.

Moreover, while recent vision-language pretrained models like BLIP-2 offer promising architectural advantages for efficient adaptation, their behaviour under language-specific tokenisation constraints and limited data conditions is not well understood. This gap motivates our study, which investigates the adaptation of a pretrained vision-language model to Vietnamese image captioning in a furniture-specific domain, with an emphasis on tokenisation strategies and generation behaviour rather than architectural modifications.

3. Methodology

This section describes the methodology adopted in this study. We first briefly review conventional image captioning approaches to establish a baseline understanding. We then present the proposed adaptation strategy based on BLIP-2 for Vietnamese furniture image captioning. Finally, we outline the overall workflow of the system.

3.1. Conventional image captioning approaches

Traditional image captioning methods typically follow an encoder-decoder paradigm [6, 21]. In this framework, a visual encoder extracts high-level features from an input image, which are subsequently used by a language decoder to generate a textual description. Early approaches employed CNNs, such

as VGG or ResNet, as visual encoders, coupled with RNNs or LSTM networks for sequence generation.

Attention mechanisms were later introduced to improve caption quality by enabling the decoder to selectively focus on relevant image regions during generation [7]. These attention-based models significantly enhanced descriptive accuracy and became the dominant paradigm prior to the emergence of Transformer-based architectures.

More recent approaches replace recurrent decoders with Transformer-based models, which leverage self-attention to better capture long-range dependencies in generated captions [10, 11]. While these methods achieve strong performance on standard benchmarks, they typically require large annotated datasets and are often trained or fine-tuned from scratch. Consequently, their effectiveness is limited in low-resource settings, particularly for languages and domains that differ significantly from the training data.

3.2. Proposed method: BLIP-2 adaptation for Vietnamese furniture captioning

In this study, we adopt BLIP-2 as the core model [12] for Vietnamese furniture image captioning due to its modular design and strong transferability. BLIP-2 consists of three main components: a frozen vision encoder (ViT-G/14, ~1.1B parameters), a frozen large language model (OPT-2.7B), and a lightweight Querying Transformer (Q-Former, ~188M parameters) that bridges the visual and textual modalities. The total model size is approximately 4B parameters.

3.2.1. Model architecture

Given an input image I , the vision encoder extracts a set of visual embeddings V . These embeddings are then queried by the Q-Former using a fixed number of learnable query tokens. The output of the Q-Former serves as a compact visual representation, which is projected into the input space of the language model. The language model subsequently generates a caption $C=(w_1, w_2, \dots, w_T)$ in an autoregressive manner.

This design allows BLIP-2 to leverage powerful pretrained components while limiting the number of trainable parameters, making it particularly suitable for domain adaptation with limited data.

3.2.2. Tokenisation strategy for Vietnamese

One key challenge in applying pretrained vision-language models to Vietnamese lies in tokenisation.

Vietnamese words are often composed of multiple syllables separated by spaces, which can be fragmented into suboptimal tokens when processed by tokenisers designed for English.

Rather than extending or retraining the tokeniser, which may introduce instability under limited data conditions, we retain the original tokeniser of the pretrained language model. This decision is motivated by the observation that modifying the tokeniser without sufficient data can negatively affect the pretrained representations. By preserving the original tokeniser, we aim to maintain the integrity of the language model while allowing the Q-Former to adapt visual representations to the existing token space. A quantitative analysis of this design decision, including a comparison with Vietnamese-optimised tokenisers, is presented in Section 4.4.

3.2.3. Fine-tuning strategy

To adapt BLIP-2 to the Vietnamese furniture domain, we fine-tune only the Q-Former and the projection layers, while keeping both the vision encoder and the language model frozen. This approach reduces the risk of overfitting and significantly lowers computational requirements.

Training is performed using BF16 precision [22] to ensure numerical stability and to mitigate gradient-related issues observed during preliminary experiments. The model is optimised using a standard cross-entropy loss over the generated tokens, conditioned on the visual input and preceding text tokens.

3.2.4. Generation configuration

During inference, we carefully adjust generation-related hyperparameters to balance descriptive richness and factual accuracy. In particular, we analyse the impact of maximum sequence length and length penalty on caption quality. Longer generation lengths can capture more detailed attributes but may also increase the risk of hallucination, while overly restrictive settings may truncate important information. These trade-offs are systematically examined in the experimental section.

3.3. System workflow

The overall workflow of the proposed approach is illustrated as follows. First, a domain-specific dataset of Vietnamese furniture images and corresponding captions is constructed through

data collection and manual annotation. Each image-caption pair is preprocessed and tokenised using the original tokeniser of the pretrained language model.

During training, input images are encoded by the frozen vision encoder, and the resulting visual features are passed to the Q-Former. The Q-Former learns to query and summarise relevant visual information, which is then aligned with the language model for caption generation. Only the Q-Former and projection layers are updated during this stage.

At inference time, a given furniture image is processed through the same pipeline to generate a Vietnamese caption. The generation process follows an autoregressive decoding strategy with carefully selected hyperparameters to ensure both fluency and factual consistency. Qualitative examples of generated captions are provided in the experimental section to illustrate the effectiveness of the proposed workflow.

4. Experiments

This section presents the experimental evaluation of the proposed approach. We first describe the experimental setup, including the dataset, evaluation metrics, and training configurations. We then analyse the model performance in comparison with multiple baselines and provide a detailed discussion of the results.

4.1. Experimental setup

4.1.1. Dataset

To evaluate the proposed method, we construct a domain-specific dataset for Vietnamese furniture image captioning. The dataset consists of 3,483 image-caption pairs collected from online furniture catalogues and e-commerce platforms. Each image is associated with a single manually curated Vietnamese caption that emphasises key product attributes such as product category, material, size, colour, and overall design style. The dataset is publicly available at <https://www.kaggle.com/datasets/minhithp/vietnamese-interior-image-captions/>.

The dataset is split into training, validation, and test sets following an approximately 80/10/10 ratio (Table 1). All images are resized and normalised according to the requirements of the pretrained vision encoder used in BLIP-2. Captions are tokenised using the original tokeniser of the pretrained language model without any modification or vocabulary expansion.

Table 1. Distribution of furniture categories in the Vietnamese image captioning dataset.

Category	Train	Val	Test	Total
Kitchen cabinets	280	35	29	344
Wardrobes	710	89	89	888
Beds	465	58	58	581
Sofa sets	240	30	30	300
TV stands	230	29	29	288
Dining table sets	160	20	20	200
Dressing tables	145	18	18	181
Shoe cabinets	145	18	18	181
Wine cabinets	95	12	12	119
Work desks	80	10	10	100
Study desks	55	7	7	69
Nightstands	40	5	5	50
Bookshelves	80	5	5	90
Bunk beds	50	2	2	54
Grandfather clocks	15	2	2	19
Filing cabinets	15	2	2	19
Total	2,805	342	336	3,483

4.1.2. Implementation details

All training and evaluation experiments were conducted on Google Colab Pro, using an NVIDIA A100-SXM4 GPU with 40 GB VRAM, 64 GB system RAM, and 500 GB SSD storage. To assess practical deployability, the inference stage was additionally verified on an NVIDIA L4 GPU (22 GB VRAM).

The model was fine-tuned using PyTorch 2.0 and Transformers 4.35, based on BLIP-2 with OPT-2.7B as the frozen language model. During training, only the Q-Former and projection layers were updated, while both the vision encoder and the language model remained frozen.

The training hyperparameters were configured as follows: a learning rate of 1×10^{-5} , an effective batch size of 32 (actual batch size of 4 with gradient accumulation of 8), and the AdamW optimiser. All experiments were performed using BF16 precision to ensure numerical stability. Training was run for a maximum of 15 epochs with early stopping (patience=4) based on validation loss.

The total training time was 174.35 minutes, and the lowest validation loss of 0.146 was achieved at epoch 12. The best-performing checkpoint was selected for inference on the test set, which consists of 336 images.

4.1.3. Evaluation metrics

We evaluate caption quality using standard automatic metrics commonly adopted in image

captioning research, including BLEU-1, BLEU-4 [23], METEOR [24], ROUGE-L [25], and CIDEr [26]. These metrics provide complementary perspectives on n-gram overlap, semantic similarity, and overall descriptive quality. In addition to quantitative evaluation, we also conduct qualitative analysis to examine linguistic fluency, attribute coverage, and hallucination behaviour.

4.1.4. Training configuration

During fine-tuning, only the Q-Former and projection layers of BLIP-2 are updated, while the vision encoder and language model remain frozen. Training is performed using BF16 precision [22] to ensure numerical stability. The model is optimised using the AdamW optimiser with a fixed learning rate. Early stopping is applied based on validation loss to prevent overfitting.

For caption generation, we employ greedy decoding and analyse the effect of different generation hyperparameters, including maximum sequence length and length penalty, on the quality of generated captions.

4.2. Comparison with baseline methods

To provide a comprehensive evaluation, we compare the fine-tuned BLIP-2 against five zero-shot baseline configurations drawn from four model families, covering diverse architectures and model scales:

GIT (Microsoft/git-base): A generative image-to-text model [27] pretrained primarily on English data, evaluated with two Vietnamese adaptation strategies: vocabulary expansion and subword tokenisation.

BLIP-2 zero-shot (Salesforce/blip2-opt-2.7b, ~4B params): The same architecture as our fine-tuned model, evaluated without any fine-tuning, to isolate the contribution of domain adaptation.

InstructBLIP (Salesforce/instructblip-flan-t5-xl, ~4.3B params): An instruction-tuned extension of BLIP-2 [14], prompted to describe furniture products in Vietnamese.

LLaVA-1.5-7B (llava-hf/llava-1.5-7b-hf, ~7.3B params): A state-of-the-art instruction-following vision-language model [13], with nearly twice the total parameters of BLIP-2, and was prompted in Vietnamese.

It is important to note that all zero-shot baselines are evaluated in inference mode on the same test set using identical evaluation metrics. LLaVA and

InstructBLIP use instruction prompts as required by their architecture, while BLIP-2 and GIT use unconditional generation, consistent with each model's designed inference paradigm.

Table 2 presents the quantitative results. All zero-shot baselines fail to generate meaningful Vietnamese captions. InstructBLIP produces exclusively English output (BLEU-4=0.00), as its Flan-T5 tokeniser loses 20.83% of Vietnamese tokens. LLaVA-1.5-7B, despite being the largest model at 7.3B parameters, either repeats the input prompt or generates generic descriptions (BLEU-4=0.02, CIDEr=0.09). BLIP-2 zero-shot simply echoes the prompt without generating descriptive content (BLEU-4=0.01).

These results demonstrate that model scale alone is insufficient for Vietnamese domain-specific captioning. In contrast, the fine-tuned BLIP-2 achieves BLEU-4 of 31.22 and CIDEr of 243.60, confirming that domain-specific adaptation with limited data is far more effective than relying on larger zero-shot models.

Table 2. Performance comparison of image captioning models on the test set.

Model	BLEU-1	BLEU-4	CIDEr	METEOR	ROUGE-L
GIT + Vocabulary expansion	4.59	0.03	3.69	3.28	18.49
GIT + Subword tokenisation	0.41	0.01	0.78	1.76	6.34
BLIP-2 OPT-2.7B (zero-shot)	0.03	0.01	0.07	6.20	19.23
InstructBLIP FlanT5-XL (zero-shot)	0.00	0.00	0.00	0.31	0.62
LLaVA-1.5-7B (zero-shot)	0.82	0.02	0.09	-	16.54
BLIP-2 (maximum length=60)	53.28	34.84	268.92	34.41	65.26
BLIP-2 (maximum length=120)	31.73	15.87	82.72	35.09	51.35
BLIP-2 Optimal (maximum length=70)	53.18	31.22	243.60	35.49	60.47
UIT-ViC	-	39.50	106.68	-	-

Models above the horizontal line are zero-shot; models below are fine-tuned on the training set.

4.3. Result analysis

The quantitative results demonstrate that the proposed BLIP-2 adaptation achieves strong performance on the Vietnamese furniture image captioning task. While the configuration with maximum length=60 attains the highest scores on several automatic metrics (including BLEU-4 and CIDEr), it frequently produces truncated captions.

In contrast, the configuration with maximum length=70 and length penalty=1.5 yields slightly lower automatic scores (BLEU-4=31.22, CIDEr=243.60) but provides a better balance between descriptive completeness and factual accuracy. In particular, this setting significantly reduces truncation while keeping hallucination at a manageable level.

As a result, we adopt maximum length=70 as the optimal configuration for subsequent analysis, as it offers more reliable and practically usable captions in a domain-specific Vietnamese setting.

An analysis across different n-gram levels shows a gradual decrease from BLEU-1 (53.18) to BLEU-4 (31.22), suggesting that the model is more reliable at recognising individual content words than generating longer and more complex sentence structures. When compared to UIT-ViIC on a general-domain dataset (BLEU-4=39.5) [16], the lower BLEU-4 score observed in our experiments is reasonable due to domain differences, smaller dataset size, and the higher level of descriptive detail required for furniture products. Notably, the proposed approach achieves a substantially higher CIDEr score (243.60) than UIT-ViIC (106.68), indicating strong consensus with human-written captions in this domain-specific setting.

We further investigate the impact of generation length on caption quality. Experiments with different maximum length values reveal a clear trade-off between truncation and hallucination. While maximum length=60 yields the highest BLEU-4 score, it leads to truncation in approximately 30% of the generated captions. Increasing the maximum length to 120 eliminates truncation but tends to introduce additional spurious attributes, further degrading factual precision. The configuration maximum length=70 combined with length penalty=1.5 achieves the best overall balance: truncation is reduced to only 1.8%, while hallucination - though it remains the dominant error type at 67.6% - is confined mainly to material and colour attributes and is kept at a level compatible with practical use. This result underscores the critical role of generation hyperparameter optimisation for achieving reliable Vietnamese image captioning in domain-specific applications.

4.4. Quantitative tokenisation analysis

To answer research question Q2, we conduct a quantitative comparison of tokenisation behaviour across three tokenisers representative of the evaluated models: OPT (used in BLIP-2), PhoBERT

(a Vietnamese-optimised tokeniser included as a reference), and Flan-T5 (used in InstructBLIP). Table 3 summarises the results on all 3,483 captions.

Table 3. Quantitative tokenisation comparison on Vietnamese furniture captions.

Tokeniser	Tok/Cap	Tok/W	OOV%	Frag%	Diac%
OPT (BLIP-2)	32.7	3.84	0.00	80.6	100.0
PhoBERT (VN)	10.1	1.17	0.00	2.5	100.0
Flan-T5	24.7	2.89	20.83	67.0	25.7

Tok/Cap=avg tokens per caption; Tok/W=avg subword tokens per Vietnamese word; OOV=out-of-vocabulary rate; Frag=words split into >2 subwords; Diac=diacritics preserved after tokenise-decode.

The OPT tokeniser fragments Vietnamese words substantially, requiring an average of 3.84 subword tokens per word compared to 1.17 for the Vietnamese-optimised PhoBERT. For example, the 5-word caption “Tủ Bếp Gỗ Công Nghiệp” is tokenised into 23 tokens by OPT but only 5 tokens by PhoBERT. This means OPT requires approximately 3.2× more tokens per caption (32.7 vs. 10.1), increasing computational cost and truncation risk. Notably, 80.6% of Vietnamese words are split into more than 2 subwords by OPT.

Despite this heavy fragmentation, OPT achieves 0% OOV rate through byte-level BPE and preserves 100% of Vietnamese diacritics after encode-decode round-trips. In contrast, the Flan-T5 tokeniser (used in InstructBLIP) exhibits a 20.83% OOV rate and preserves only 25.7% of diacritics, directly explaining InstructBLIP’s complete failure in Vietnamese caption generation (Table 2).

We also investigated whether Vietnamese word segmentation could reduce OPT’s fragmentation. Using the Underthesea segmenter to join compound words, OPT tokenisation actually worsened (45.2 tokens/caption vs. 40.4 for raw text), as underscore connectors introduced additional fragmentation. These findings justify our decision to retain the original OPT tokeniser: accepting higher tokenisation fragmentation to preserve the pretrained language model’s knowledge, while relying on the Q-Former to learn effective mappings from fragmented token sequences to visual features.

4.5. Error analysis

To provide a systematic understanding of model limitations, we conduct an automatic error classification on all 336 test captions generated by the optimal BLIP-2 configuration. Table 4 presents the results.

Table 4. Error classification of generated captions on the test set (336 images).

Error type	Count	Rate (%)
No error	96	28.6
Hallucinated material	166	49.4
Hallucinated colour	95	28.3
Hallucinated dimension	60	17.9
Category mismatch	15	4.5
Truncation	6	1.8
Missing diacritics	2	0.6
Overall hallucination	227	67.6
Mean attribute overlap	-	65.0

Hallucination is the most prevalent error type, with 67.6% of captions containing at least one hallucinated attribute. The most frequently hallucinated attributes are materials (e.g., “MDF” appearing 43 times, “Melamine” 33 times) and colours (e.g., “white” 47 times, “grey” 46 times). These hallucinations reflect the model’s tendency to generate statistically frequent domain attributes that are not present in the specific image.

Performance varies substantially across furniture categories. Dining table sets achieve the highest no-error rate (68.4%), likely due to their visually distinctive appearance. In contrast, dressing tables (5.0%) and work desks (0.0%) are most challenging, possibly due to greater visual diversity and fewer training samples.

Category mismatch (4.5%) and truncation (1.8%) are relatively rare under the optimal configuration, confirming the effectiveness of generation hyperparameter tuning. Missing diacritics occur in only 0.6% of cases, consistent with the tokenisation analysis showing 100% diacritics preservation by the OPT tokeniser.

4.6. Hyperparameter sensitivity analysis

To examine the sensitivity of model performance to training hyperparameters, we conduct sensitivity experiments varying learning rate and effective batch size. Due to computational constraints, each configuration is trained for 3 epochs to observe convergence trends. Table 5 presents the results.

Table 5. Sensitivity analysis on training hyperparameters (3 epochs).

Configuration	LR	Batch	BLEU-4	CIDEr
Baseline	1e-5	32	20.51	146.39
Higher LR	5e-5	32	22.50	173.31
Lower LR	5e-6	32	11.57	61.03
Smaller batch	1e-5	8	26.39	205.70

The results reveal clear sensitivity to both learning rate and batch size. A learning rate of 5×10^{-6} leads to substantially slower convergence (CIDEr=61.03 at 3 epochs), while 5×10^{-5} provides marginally faster early convergence. The most notable finding is that a smaller effective batch size (8) achieves the best early-stage performance (CIDEr=205.70), as it provides $4 \times$ more frequent parameter updates.

However, the baseline configuration (LR= 10^{-5} , batch=32) trained for the full 15 epochs with early stopping achieves the best final performance (CIDEr=243.60), indicating that larger batch sizes benefit from extended training with more stable gradient estimates. These findings validate our selection of training hyperparameters.

4.7. Qualitative results

Figure 1 presents qualitative examples of the generated captions for representative furniture images. The model accurately identifies product types, materials, and key attributes in most cases, demonstrating effective domain adaptation.

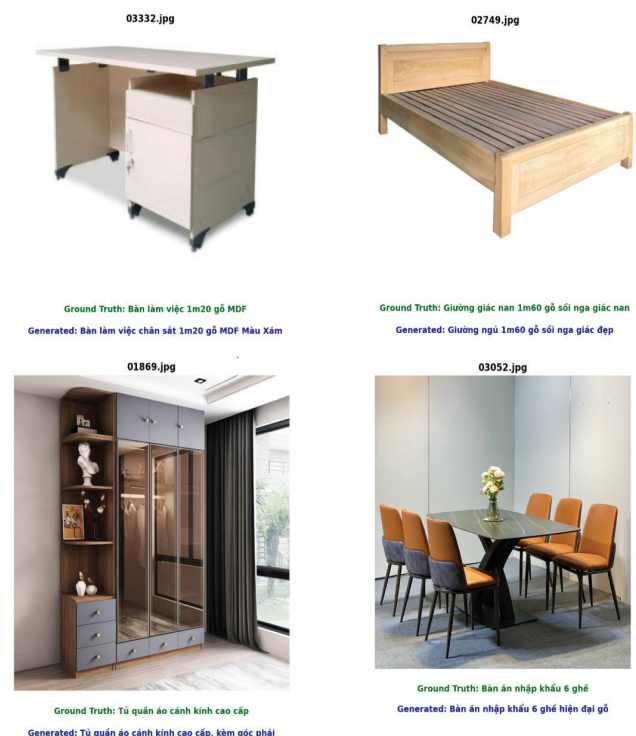


Fig. 1. Qualitative results of the proposed Vietnamese image captioning model in the furniture domain. For each example, the ground-truth caption (in green) is compared with the caption generated by the model (in blue). The results demonstrate that the model is able to accurately capture key attributes of interior furniture items.

5. Conclusions

In this article, we investigated the problem of Vietnamese image captioning in a domain-specific furniture setting, where both linguistic characteristics and domain requirements pose significant challenges to existing vision-language models. To address these challenges, we constructed a dedicated Vietnamese furniture image captioning dataset (publicly available at Kaggle) and studied the adaptation of a pretrained vision-language model, BLIP-2, under limited data conditions.

Comprehensive experiments demonstrate the necessity and effectiveness of domain-specific fine-tuning. All five zero-shot baselines, including LLaVA-1.5-7B with 7.3B parameters, fail to generate meaningful Vietnamese captions (BLEU-4 $\leq$ 0.03), while the fine-tuned BLIP-2 achieves BLEU-4 of 31.22 and CIDEr of 243.60.

Quantitative tokenisation analysis reveals that the OPT tokeniser fragments Vietnamese words into an average of 3.84 subwords (vs. 1.17 for PhoBERT), yet preserves all diacritics with 0% OOV rate, justifying the decision to retain the original tokeniser. Systematic error analysis identifies hallucination as the primary challenge (67.6% of captions), particularly for materials and colours, while truncation (1.8%) and missing diacritics (0.6%) are minimal under the optimal configuration. The sensitivity analysis validates the impact of learning rate and batch size on model performance, with the selected hyperparameters (LR=10⁻⁵, batch=32, 15 epochs) achieving the best overall results.

Our analysis further highlights the importance of preserving the original tokeniser of the pretrained language model when adapting vision-language systems to Vietnamese. In contrast, English-centred models without language-specific adaptation fail to generate coherent Vietnamese captions, providing empirical evidence of the critical role played by tokenisation and pretrained language representations. In addition, experiments on generation hyperparameters reveal a clear trade-off between truncation and hallucination, emphasising that careful tuning of decoding strategies is essential for achieving reliable and factual image captions in specialised domains.

Despite these promising results, several limitations remain. First, the dataset size is relatively modest, and each image is associated with only a single reference caption, which may

limit the robustness of automatic evaluation metrics. Second, although the proposed approach reduces hallucination compared to naive adaptation strategies, occasional inaccuracies persist, particularly for visually subtle attributes such as material texture or fine-grained design details. Finally, extending the evaluation to include fine-tuned versions of LLaVA and InstructBLIP on the same dataset could provide additional insights into cross-architecture comparison.

Future research directions include expanding the dataset with multiple reference captions per image to enhance linguistic diversity and evaluation reliability. Incorporating additional visual grounding mechanisms or auxiliary tasks, such as object or attribute detection [28, 29], may further reduce hallucination and improve factual consistency. Moreover, extending the analysis to multilingual and instruction-tuned vision-language models could provide deeper insights into the generalisation capabilities of vision-language systems for low-resource languages and domain-specific applications. Overall, this work provides a solid foundation for both academic research and practical deployment of Vietnamese image captioning systems in real-world e-commerce environments.

CRedit author statement

Cong Minh Vu: Conceptualisation, Methodology, Investigation, Writing original draft; Quang Hung Nguyen: Supervision, Methodology Guidance, Writing - Reviewing & Editing.

ACKNOWLEDGEMENTS

The research was supported by the project 12-2026-HV-CNTT1 funded by the Posts and Telecommunications Institute of Technology (PTIT).

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] J. Li, D. Li, C. Xiong, et al. (2022), "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation", *Computer Vision and Pattern Recognition*, **162**, pp.12888-12900, DOI: 10.48550/arXiv.2201.12086.
- [2] A. Radford, J.W. Kim, C. Hallacy, et al. (2021), "Learning transferable visual models from natural language supervision", *Computer Vision and Pattern Recognition*, **139**, pp.8748-8763, DOI: 10.48550/arXiv.2103.00020.

- [3] P. Wang, A. Yang, R. Men, et al. (2022), "OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework", *Computer Vision and Pattern Recognition*, **162**, pp.23318-23340, DOI: 10.48550/arXiv.2202.03052.
- [4] D.Q. Nguyen, T. Vu, D.Q. Nguyen, et al. (2017), "From word segmentation to POS tagging for Vietnamese", *Proceedings of The Australasian Language Technology Association Workshop 2017*, pp.108-113.
- [5] K.V. Nguyen, D.V. Nguyen, A.G.T. Nguyen, et al. (2020), "A Vietnamese dataset for evaluating machine reading comprehension", *Proceedings of The 28th International Conference on Computational Linguistics*, pp.2595-2605, DOI: 10.18653/v1/2020.coling-main.233.
- [6] O. Vinyals, A. Toshev, S. Bengio, et al. (2015), "Show and tell: A neural image caption generator", *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pp.3156-3164, DOI: 10.1109/CVPR.2015.7298935.
- [7] K. Xu, J. Ba, R. Kiros, et al. (2015), "Show, attend and tell: Neural image caption generation with visual attention", *Machine Learning*, **37**, pp.2048-2057, DOI: 10.48550/arXiv.1502.03044.
- [8] P. Anderson, X. He, C. Buehler, et al. (2018), "Bottom-up and top-down attention for image captioning and visual question answering", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.6077-6086, DOI: 10.1109/CVPR.2018.00636.
- [9] A. Vaswani, N. Shazeer, N. Parmar, et al. (2017), "Attention is all you need", *Advances in Neural Information Processing Systems*, **30**, pp.5998-6008.
- [10] M. Cornia, M. Stefanini, L. Baraldi, et al. (2020), "Meshed-memory transformer for image captioning", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.10578-10587, DOI: 10.1109/CVPR42600.2020.01059.
- [11] Y. Pan, T. Yao, Y. Li, et al. (2020), "X-linear attention networks for image captioning", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.10968-10977, DOI: 10.1109/CVPR42600.2020.01098.
- [12] J. Li, D. Li, S. Savarese, et al. (2023), "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models", *Computer Vision and Pattern Recognition*, **202**, pp.19730-19742, DOI: 10.48550/arXiv.2301.12597.
- [13] H. Liu, C. Li, Q. Wu, et al. (2023), "Visual instruction tuning", *Advances in Neural Information Processing Systems*, **36**, pp.34892-34916.
- [14] W. Dai, J. Li, D. Li, et al. (2023), "InstructBLIP: Towards general-purpose vision-language models with instruction tuning", *Advances in Neural Information Processing Systems*, **36**, pp.49250-49267.
- [15] X. Zhao, Z. Yang, Q. Li, et al. (2025), "Parameter-efficient transfer learning for remote sensing image captioning", *IEEE Transactions on Geoscience and Remote Sensing*, **63**, DOI: 10.1109/TGRS.2025.3584887.
- [16] Q.H. Lam, Q.D. Le, K.V. Nguyen, et al. (2020), "UIT-ViC: A dataset for the first evaluation on Vietnamese image captioning", *Computational Collective Intelligence: 12th International Conference (ICCCI 2020), Lecture Notes in Computer Science*, pp.730-742, DOI: 10.1007/978-3-030-63007-2_57.
- [17] K.D. Chowdhury, M. Hasanuzzaman, Q. Liu (2018), "Multimodal neural machine translation for low-resource language pairs using synthetic data", *Proceedings of The Workshop on Deep Learning Approaches for Low-Resource NLP*, pp.33-42, DOI: 10.18653/v1/W18-3405.
- [18] B. Jing, P. Xie, E. Xing (2018), "On the automatic generation of medical imaging reports", *Proceedings of The 56th Annual Meeting of The Association for Computational Linguistics (Volume 1: Long Papers)*, pp.2577-2586, DOI: 10.18653/v1/P18-1240.
- [19] A. Salvador, N. Hynes, Y. Aytar, et al. (2017), "Learning cross-modal embeddings for cooking recipes and food images", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.3068-3076, DOI: 10.1109/CVPR.2017.327.
- [20] S. Vittayakorn, K. Yamaguchi, A. C. Berg, et al. (2015), "Runway to realway: Visual analysis of fashion", *Proceedings of The IEEE Winter Conference on Applications of Computer Vision*, pp.951-958, DOI: 10.1109/WACV.2015.131.
- [21] K. Cho, B.V. Merriënboer, C. Gulcehre, et al. (2014), "Learning phrase representations using RNN encoder-decoder for statistical machine translation", *Proceedings of The 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1724-1734, DOI: 10.3115/v1/D14-1179.
- [22] P. Micikevicius, S. Narang, J. Alben, et al. (2018), "Mixed precision training", *Proceedings of The International Conference on Learning Representations (ICLR 2018)*, DOI: 10.48550/arXiv.1710.03740.
- [23] K. Papineni, S. Roukos, T. Ward, et al. (2002), "BLEU: A method for automatic evaluation of machine translation", *Proceedings of The 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp.311-318, DOI: 10.3115/1073083.1073135.
- [24] S. Banerjee, A. Lavie (2005), "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments", *Proceedings of The ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp.65-72.
- [25] C.Y. Lin (2004), "ROUGE: A package for automatic evaluation of summaries", *Text Summarization Branches Out*, pp.74-81.
- [26] R. Vedantam, C.L. Zitnick, D. Parikh (2015), "CIDEr: Consensus-based image description evaluation", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.4566-4575, DOI: 10.1109/CVPR.2015.7299087.
- [27] J. Wang, Z. Yang, X. Hu, et al. (2022), "GIT: A generative image-to-text transformer for vision and language", *Transactions on Machine Learning Research*, DOI: 10.48550/arXiv.2205.14100.
- [28] T. Yao, Y. Pan, Y. Li, et al. (2018), "Exploring visual relationship for image captioning", *Proceedings of The European Conference on Computer Vision, Lecture Notes in Computer Science*, pp.684-699, DOI: 10.1007/978-3-030-01264-9_42.
- [29] Q. Wu, C. Shen, L. Liu, et al. (2016), "What value do explicit high level concepts have in vision to language problems?", *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.203-212, DOI: 10.1109/CVPR.2016.29.

MDToC: Metacognitive Dynamic Tree of Concepts for boosting mathematical problem-solving of large language models

Tung Duong Ta^{1*}, Tim Oates¹, Thien Van Luong², Huan Vu², Tien Cuong Nguyen³

¹Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, Maryland 21250, USA

²National Economics University, 207 Giai Phong Road, Bach Mai Ward, Hanoi, Vietnam

³VNPT AI, Lane 55 Huynh Thuc Khang Street, Yen Hoa Ward, Hanoi, Vietnam

Received 2 July 2026; revised 31 August 2026; accepted 4 September 2026

Abstract:

Despite advances in mathematical reasoning capabilities, Large Language Models (LLMs) still struggle with calculation verification when using established prompting techniques. Chain-of-Thought prompting and its self-consistency variant lack evaluation of intermediate reasoning steps. Hierarchical techniques such as Tree-of-Thoughts (ToT) and Graph-of-Thoughts (GoT) evaluate abstract thoughts, which are difficult to score reliably. We present MDToC (Metacognitive Dynamic Tree of Concepts), a three-phase prompting technique built on the metacognitive stages of planning, monitoring, and reviewing. MDToC constructs a concept tree, develops accuracy-verified calculations for each concept, and employs majority voting to evaluate competing solutions. Evaluations across CHAMP, MATH, and Game-of-24 benchmarks demonstrate our MDToC's effectiveness, with GPT-4-Turbo achieving 58.1% on CHAMP, 86.6% on MATH, and 85% on Game-of-24, outperforming GoT by 5%, 5.4%, and 4% on all these tasks, respectively, without hand-engineered hints. MDToC consistently matches or surpasses existing prompting methods across all backbone models, yielding improvements of up to 19% over ToT and 13% over GoT. In addition, delegating auxiliary components to lighter models reduces cost while maintaining accuracy. These results establish metacognitive calculation verification as a promising direction for enhanced mathematical reasoning.

Keywords: large language models, metacognition, mathematical reasoning, problem-solving, tree of concepts.

1. Introduction

Large Language Models (LLMs) like GPT-4 [1] and Claude [2] demonstrate proficiency in various mathematical problems, excelling in easy to medium-difficulty tasks as evidenced by their performance on benchmarks such as GSM8k [3] and SVAMP [4]. However, their efficacy diminishes when faced with complex challenges presented in datasets such as MATH [5] and CHAMP [6]. In these demanding scenarios, LLMs often struggle with accurate multi-step reasoning and solution derivation. A key factor in this performance degradation is the models' propensity for errors in intermediate calculations and logical deductions [7, 8]. These compounding inaccuracies result in poor

performance on datasets featuring hard mathematical problems, opening a critical area for improvement in the multi-step reasoning capabilities of LLMs. Recent surveys of mathematical reasoning with LLMs likewise identify the verification of intermediate calculations as a central open challenge [9].

Researchers have widely adopted prompting techniques, particularly Chain-of-Thought (CoT) [10] and self-consistency CoT (SC-CoT) [11], to enhance LLMs' multi-step reasoning capabilities without additional training. These methods enable models to decompose complex reasoning processes into smaller steps, improving overall accuracy. In particular, CoT encourages articulation of thought processes, while SC-CoT generates multiple demonstrations with majority voting. However, these approaches have limitations: CoT may constrain diverse

*Corresponding author: Email: dta1@umbc.edu

problem-solving pathways, while SC-CoT lacks crucial evaluation of intermediate reasoning steps. This can lead to erroneous samples and inaccurate voting outcomes. As a result, using these prompting techniques, advanced LLMs such as GPT-4 suffer from poor performance. For example, GPT-4 with SC-CoT achieves only 9% accuracy on the Game-of-24 task [12]. Therefore, it is essential to develop more robust reasoning methodologies.

Recent hierarchical prompting techniques like Tree-of-Thoughts (ToT) [12, 13] and Graph-of-Thoughts (GoT) [14, 15] have advanced reasoning capabilities through structured thought representation and intermediate evaluation, achieving impressive results on complex tasks (74% accuracy on Game-of-24 [12] and 89% on Sequence-Sorting-64-elements [14], respectively). However, as shown in Fig. 1, these approaches suffer from ill-defined evaluation criteria for diverse thought forms (mathematical analysis, concepts, calculations), leading to heavy reliance on powerful LLMs like GPT-4 that produce approximated and unreliable evaluation scores. This standardization challenge creates vulnerabilities in the critical processes of thought selection and connection pruning, while the need for domain-specific customization limits generalizability across different problem types.

Amidst the numerous successes of the CoT, ToT, and GoT prompting techniques, there have been several explorations of cognitive prompting methods for mathematical problem-solving [16]. In the field of psychology, metacognition enables individuals to reflect on and critically analyze their thought processes [17]. Recent research has enhanced model capabilities with metacognitive processes for natural language understanding tasks. For example, Y. Wang, et al. (2023) [18] demonstrated that LLMs prompted with metacognitive thinking outperformed previous techniques such as zero-shot [19, 20] or CoT prompting [10] across various NLP tasks. Y. Zhou, et al. (2024) [21] highlighted the effectiveness of the metacognitive approach in improving the retrieval-augmented generation process for LLMs. However, the application of metacognition to mathematical problem-solving remains relatively unexplored, with notable exceptions such as A. Didolkar, et al. (2024) [22], who showed that metacognitive approaches enhance mathematical reasoning in LLMs by reflecting on clustered math skills and thereby providing relevant in-context examples. Our work extends this research direction by applying metacognition to LLMs for solving mathematical problems.

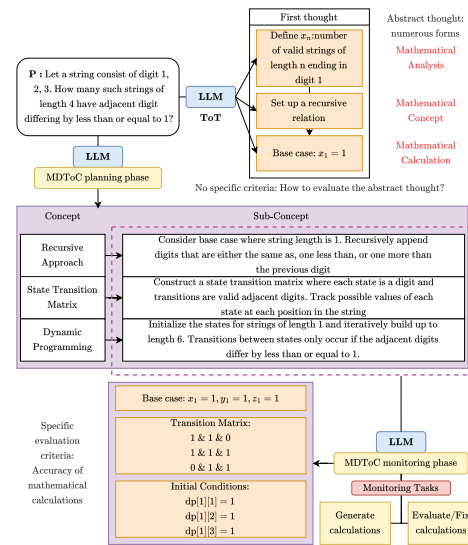


Fig. 1. ToT prompting yields initial abstract thoughts (e.g., analyses, concepts, calculations; in red), which are challenging to evaluate due to the intangible nature of conceptual reasoning and the lack of specific criteria to measure their correctness or completeness. Our MDToC addresses these abstract thoughts by first generating concrete concepts and then producing relevant calculations for those concepts. We only evaluate the preciseness of the calculations through mathematical accuracy checks, enabling precise evaluation and thus improving problem-solving reliability.

Motivated by the aforementioned background, we propose MDToC (Metacognitive Dynamic Tree of Concepts), a novel three-phase prompting technique that transforms abstract thoughts into concepts and evaluable calculations. MDToC employs a depth-two concept tree in the planning phase to explore diverse mathematical concepts while constraining the solution space, followed by a monitoring phase that expands sub-concepts with calculation steps using four specialized LLMs, and concludes with a reviewing phase utilizing majority voting following the self-consistency voting mechanism [23]. This comprehensive framework has demonstrated significant effectiveness, with GPT-3.5+MDToC achieving 39.5% accuracy on CHAMP (outperforming GPT-3.5 with annotated concepts by 5.1%) and GPT-4o-mini+MDToC attaining 75% accuracy on Game-of-24 (surpassing GPT-4o-mini with ToT by 19%).

2. Materials and methods

This section first reviews common prompting techniques and related work on self-feedback and training-based reasoning. It then describes the metacognitive foundation and the proposed MDToC.

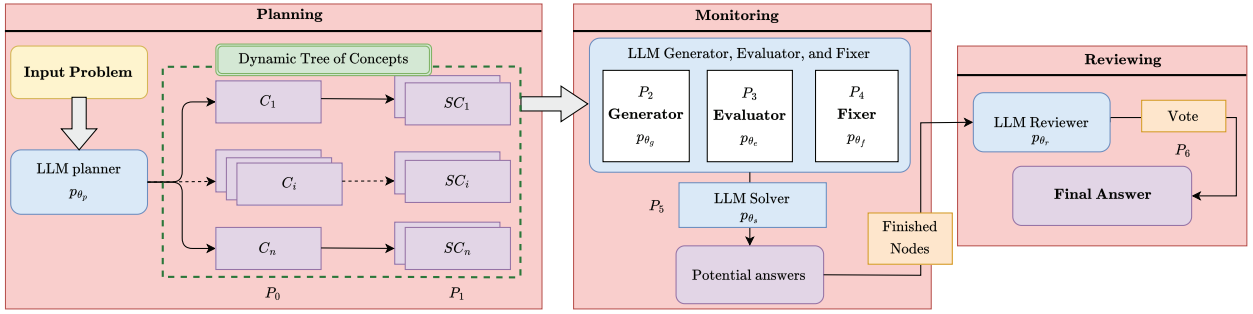


Fig. 2. Proposed MDToC prompting structure. C represents the first-depth concept, while SC represents the second-depth sub-concept. P_0 and P_1 are prompts used in the planning phase shown in Fig. 3, while P_2 , P_3 , P_4 , and P_5 are prompts used in the monitoring phase given in Fig. 4. Prompt P_6 is used in the reviewing phase, as shown in Fig. 6.

2.1. Prompting techniques

Let p_θ be an LLM parameterized by θ . Three of the most common prompting techniques for mathematical reasoning with p_θ are described below.

1. **Chain-of-Thought.** Given an input problem x , the LLM is guided through a sequence c of reasoning steps to produce an answer y [10]. Specifically, the sequence c of reasoning steps is generated by the LLM based on the input problem, expressed as $c \sim p_\theta(c|x)$. The final answer y is then generated by the LLM based on both the input problem and the sequence of reasoning steps expressed as $y \sim p_\theta(y|x, c)$.
2. **Tree-of-Thought.** Given an input problem x , the LLM navigates a tree structure where each node i represents a state $s = [x, z_{1...i}]$, with $z_{1...i}$ denoting a sequence of thoughts along the current path [12]. The LLM generates a new thought z_{i+1} based on the current state s , expressed as $z_{i+1} \sim p_\theta(z_{i+1}|x, z_{1...i})$. A new node with the state $s' = [x, z_{1...i+1}]$ is then appended to the current node i in the tree. Each state s in a set of states S undergoes evaluation through either numerical values or voting to determine the viability of further path exploration. The numerical evaluation $V(p_\theta, s)$ is expressed as $V(p_\theta, s) \sim p_\theta(v|s) \forall s \in S$, where v is the numerical value. The voting evaluation $V(p_\theta, s)$ is expressed as $V(p_\theta, s) = 1 [s = s^*]$ where $s^* \sim p_\theta^{vote}(s^*|S)$. In this context, the LLM votes for state s^* given the set of states S , employing the indicator function $1 [s = s^*]$ to determine whether a state s corresponds to the voted state s^* .
3. **Graph-of-Thought** Given an input problem x , the LLM navigates a directed graph structure $G = (V, E)$, where V is the set of vertices representing thoughts, and $E \subseteq V \times V$ is the set of edges representing dependencies

among thoughts [14]. Denote that V^+ and E^+ represent newly added vertices and edges, while V^- and E^- denote removed vertices and edges, respectively. Unless stated otherwise, $V^- = E^- = \emptyset$. The graph of thoughts is manipulated through three primary operations: aggregation, refinement, and generation.

Following these operations, the graph is updated as $G' = T(G, p_\theta) = (V', E')$, where $V' = (V \cup V^+) \setminus V^-$, $E' = (E \cup E^+) \setminus E^-$. Each node v in graph G is subsequently evaluated by the LLM using either a scoring or ranking method. The scoring function is expressed as $s = E(p_\theta, v, G) \sim p_\theta(s|v, G)$, where s denotes the score value of node v . Conversely, the ranking function is expressed as $\{v_1, v_2, \dots, v_h\} = R(p_\theta, h, G) \sim p_\theta(\{v_1, v_2, \dots, v_h\}|G, h)$, where h represents the number of top-ranking thoughts to be returned.

2.2. Iterative self-feedback and training-based Reasoning

Beyond hierarchical prompting techniques, a related line of work improves reasoning through iterative self-feedback. Self-Refine [24] prompts a single LLM to evaluate and refine its own response repeatedly, but the evaluation is applied to the whole response rather than to intermediate calculations. In contrast, the monitoring phase of our MDToC uses LLMs to assess and correct only the intermediate calculations. Furthermore, it limits repeated evaluation and regeneration to two attempts per calculation. Recent findings support this design. Self-correction by LLMs without external feedback is unreliable [25]. Moreover, LLMs frequently re-evaluate calculations that are already correct. This increases token consumption without detecting new errors [26].

Whereas these methods, like MDToC, operate at inference time, another line of work improves

mathematical reasoning through training. For example, DeepSeekMath [27] continues pretraining open-source LLMs on mathematical data and applies policy optimization. More recently, a verifier has been trained to produce feedback for each reasoning step [28]. Surveys document rapid progress in fine-tuning and reinforcement learning for mathematical reasoning [9]. Our MDToC requires no additional training and can be applied to any backbone model, including such trained LLMs.

2.3. Metacognition

Metacognition, the ability to reflect on and regulate one's thought processes, plays a crucial role in advanced problem-solving and decision-making. It serves as an overarching framework guiding the effective application of cognitive strategies. This study aims to endow language models with a simulated metacognitive process, mimicking the human capacity for "thinking about thinking". Our MDToC approach employs a hierarchical prompting structure incorporating three foundational stages of metacognition: planning, monitoring, and reviewing [29], specifically designed for mathematical problem-solving.

The planning stage creates a conceptual roadmap by establishing strategies and approaches. During monitoring, we implement a metacognitive mechanism that enables self-evaluation and correction of calculations in progress. The final reviewing stage examines solutions, filtering out empty results and identifying the most frequently occurring valid answer.

2.4. The proposed MDToC

This research introduces a novel prompting approach, called MDToC, utilizing a dynamic tree of concepts within a tripartite metacognitive framework of planning, monitoring, and reviewing. This method addresses limitations in existing hierarchical prompting techniques for LLMs, such as unreliable evaluations of abstract thoughts and lack of generalizability. Our approach enables the exploration of diverse reasoning paths and selects the final solution through majority voting. Fig. 2 shows a visual representation of our methodology.

2.4.1. Planning

During the initial planning phase, we construct a concept tree $T = (V, E)$ with a depth of two where V is a set of nodes and E is a set of edges. We begin by instructing our LLM planner p_{θ_p} with prompt P_0 to extract the objective of

Prompt 0 (P_0)	Generate a minimum of $C_{\min}$ and a maximum of $C_{\max}$ distinct problem-specific concepts
Prompt 1 (P_1)	Extract all problem-solving information. Then generate a minimum of $SC_{\min}$ and a maximum of $SC_{\max}$ distinct sub-concepts filled with the extracted information given an explored concept

Fig. 3. Prompts for the planning phase.

the question, called q , and generate n distinct concepts $\{c_1^{d=1}, c_2^{d=1}, \dots, c_n^{d=1}\} \sim p_{\theta_p}(q)$, where $d = 1$ represents the first depth of the tree. Each i -th concept is articulated as a detailed sentence, providing a mathematical or programmatic response to q . We then incorporate n concepts as nodes within the tree structure, where $V = \{c_1^{d=1}, c_2^{d=1}, \dots, c_n^{d=1}\}$.

For each i -th concept at depth $d = 1$, we further prompt our LLM planner p_{θ_p} with prompt P_1 to produce m distinct sub-concepts $\{c_{i1}^{d=2}, c_{i2}^{d=2}, \dots, c_{im}^{d=2}\} \sim p_{\theta_p}(q, c_i^{d=1})$. These sub-concepts, each expressed in two detailed sentences, serve to elucidate and expand upon the i -th concept with question-solving information. Within our tree structure, we position each j -th sub-concept as a child node to its corresponding i -th concept. Fig. 3 demonstrates our prompts for P_0 and P_1 . Our tree T is then expressed as:

$$V = \{c_1^{d=1}, \dots, c_n^{d=1}, c_{11}^{d=2}, \dots, c_{1m}^{d=2}, \dots, c_{n1}^{d=2}, \dots, c_{nm}^{d=2}\}$$

$$E = \{(c_1^{d=1}, c_{11}^{d=2}), \dots, (c_n^{d=1}, c_{nm}^{d=2})\}$$

The construction of our dynamic tree T incorporates four key parameters: $C_{\min}$, $C_{\max}$, $SC_{\min}$, and $SC_{\max}$. These parameters define the concept range ($C_{\min}$ and $C_{\max}$) and sub-concept range ($SC_{\min}$ and $SC_{\max}$), as illustrated in prompts P_0 and P_1 of Fig. 3, respectively. The magnitude of these parameters directly correlates with the breadth of concept exploration, where larger values facilitate a more extensive conceptual landscape. Upon examining the annotated concepts in CHAMP [6] where each problem receives 3 hints, we decided to opt for a two-depth dynamic tree structure. We aim to prevent LLMs from excessively relying on generated concepts while fostering a more flexible problem-solving process that can adapt to various problem types and complexities.

2.4.2. Monitoring

In the monitoring phase, we employ a ToT-based structure [12] to solve a concept tree in t iterations. We denote a mathematical calculation as χ . Unlike the traditional ToT approach which samples k thoughts and selects a subset of them, our MDToC samples and evaluates all k mathematical calculations with two LLM components: an LLM evaluator p_{θ_e} and

Prompt 2 (P_2)	Derive the next small calculation step to deduce the partial solution. Any unknown computation not in the solution is not allowed for the deduction. Do not generate explanation texts of the step.
Prompt 3 (P_3)	Verify if the next calculation is correctly computed. Use code to confirm your verification. Only verify the accuracy.
Prompt 4 (P_4)	Recalculate the next calculation step-by-step. Verify if the output of the recalculation matches the original calculation. If not match, fix the output!
Prompt 5 (P_5)	Given the partial solution, continue solving and derive the final answer!

Fig. 4. Prompts for the monitoring phase.

an LLM generator p_{θ_g} . The generator p_{θ_g} prompted with P_2 samples k calculations, while the evaluator p_{θ_e} prompted with P_3 assesses the accuracy of each calculation. Specifically, when the evaluator p_{θ_e} identifies an error in a calculation, it returns a negative response (“No”) accompanied by a detailed explanation of the error.

The k -th calculation $\chi_{ijk}^{d \geq 3}$ is sampled from the generator as $\chi_{ijk}^{d \geq 3} \sim p_{\theta_g}(\chi_{ijk}^{d \geq 3} | c_i^{d=1}, c_{ij}^{d=2}, \chi_{ijk}^{d-1 \geq 3})$ where $\chi_{ijk}^{d-1 \geq 3}$ represents previous calculations. The evaluation result $V(p_{\theta_e}, \chi_{ijk}^{d \geq 3})$ for the k -th calculation is expressed as $V(p_{\theta_e}, \chi_{ijk}^{d \geq 3}) \sim p_{\theta_e}(v_e, r_e | (c_i^{d=1}, c_{ij}^{d=2}, \chi_{ijk}^{d-1 \geq 3}, \chi_{ijk}^{d \geq 3}))$, where v_e is the binary result of 0 or 1 and r_e is the evaluation reason when $v_e = 0$. If $v_e = 0$, the generator regenerates the calculation $\chi_{ijk}^{d \geq 3} \sim p_{\theta_g}(\chi_{ijk}^{d \geq 3} | c_i^{d=1}, c_{ij}^{d=2}, \chi_{ijk}^{d-1 \geq 3}, \chi_{ijk}^{d \geq 3}, r_e)$.

We further introduce an LLM fixer p_{θ_f} prompted with P_5 to fix the errors in the k -th calculation as $\chi_{ijk}^{d \geq 3} \sim p_{\theta_f}(\chi_{ijk}^{d \geq 3} | \chi_{ijk}^{d \geq 3})$. Specifically, after the complete cycle of generation and evaluation of calculations, p_{θ_f} addresses and corrects any remaining calculation errors. Subsequently, these k calculations are appended as nodes to the tree. After some iterations, we treat the current series of calculations as a partial solution and employ an LLM solver p_{θ_s} prompted with P_4 to resolve the partial solution. The solver’s response is subsequently appended to the tree and marked as a “Finished Node”, thereby terminating the exploratory process. Fig. 4 illustrates our prompts P_2, P_3, P_4 , and P_5 .

To monitor our concept tree with calculations, we introduce two additional parameters: c_s and t . c_s specifies the number of intermediate calculations generated at each iteration, while t represents the number of iterations. These parameters allow us to control the exploration behavior, balancing between breadth and depth.

Example analysis. Fig. 5 compares intermediate reasoning steps between GPT-3.5-Turbo with CoT + CH and GPT-3.5-Turbo with MDToC. It highlights challenges in recursive calculations requiring

precise intermediate computations. Fig. 5A shows GPT-3.5-Turbo’s performance is limited to accurate calculation of the base case only, with errors emerging in y_2 and cascading through subsequent steps. This leads to an incorrect final sum of 34.

Conversely, Fig. 5B illustrates the efficacy of our MDToC approach in mitigating computational errors. When applying MDToC, the likelihood of erroneous intermediate calculations is significantly reduced with 2 evaluation and regeneration attempts. This is evidenced by the correct computation of y_2 , accurately determined as the sum of $x_1 = 1$, $y_1 = 1$, and $z_1 = 1$, yielding the correct result of 3. This precise evaluation of x_2, y_2 , and z_2 serves as a crucial foundation, enabling accurate subsequent calculations for strings of lengths of 3 and 4, ultimately leading to the correct final answer of 41.

2.4.3. Reviewing

In the last phase, we obtain the results from the monitored tree and select the top-voted answers. Given a list of solutions marked by “Finished Node” as A , we use an LLM reviewer p_{θ_r} prompted with P_6 shown in Fig. 6 to conduct a majority voting of A . This stage involves finding the most common answer from A , returned as $\tilde{a} \sim p_{\theta_r}(A)$, where $\tilde{a}$ is the final solution to the objective q .

3. Results and discussion

3.1. Dataset

Our MDToC approach is evaluated using three datasets: CHAMP [6] (270 high school-level competition math problems across five categories, providing insight into concept tree-driven reasoning), MATH [5] (12,500 competition-level problems from sources like AIME [30] and AMC 10/12 [31] covering various advanced topics including algebra, geometry, and number theory), and a subset of 100 challenging Game-of-24 puzzles (indexed 901-1000, selected for direct comparison with ToT approaches) [12].

3.2. Parameters

Planning phase For the CHAMP and MATH datasets, we set $C_{min} = 2, C_{max} = 4, SC_{min} = 1$, and $SC_{max} = 2$ (see Section 2.4.1). In contrast, all parameters were set to 1 for Game-of-24 experiments, as these problems require diverse calculation combinations rather than varied mathematical concepts.

Monitoring phase For the CHAMP and MATH datasets, we use $c_s = 2$ and $t = 10$ (see Section 2.4.2), emphasizing in-depth concept decomposition

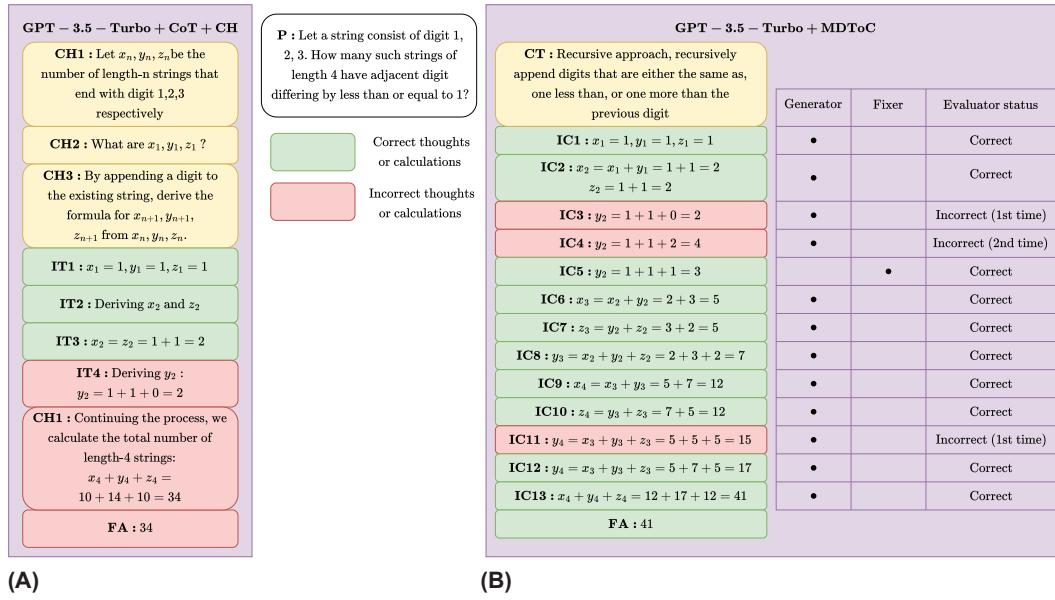


Fig. 5. Comparative analysis of reasoning steps: GPT-3.5-Turbo with CoT and CH versus GPT-3.5-Turbo with our MDToC approach. Subfigure (A) displays GPT-3.5-Turbo's reasoning with CoT, supplemented by annotated concepts and hints (CH) intended to guide the model's step-by-step reasoning; IT denotes intermediate thoughts, and FA indicates the final answer. Although these conceptual hints attempt to structure the problem-solving process, GPT-3.5-Turbo still yields an incorrect count of **34**. Because there is no automatic mechanism to spot and correct mistakes in intermediate steps, the model's calculation errors persist through to the final answer. Subfigure (B) shows our proposed concept tree (CT) approach under a multi-attempt evaluator and fixer framework, referred to here as MDToC. IC stands for intermediate calculations, and each is evaluated by an evaluator component. In this example, **IC3** and **IC4** are identified as incorrect, triggering the fixer to regenerate corrected values in **IC5**. This iterative refine-and-fix process avoids propagating calculation errors, ultimately yielding the correct final answer **FA** of **41**. Notably, this process requires no extra annotated hints, only the concept tree plus repeated evaluation up to 2 attempts, a threshold chosen to reduce the risk of model "hallucinations" (erroneous or fabricated steps).



Fig. 6. Prompts for the reviewing phase.

and analysis. Game-of-24 employs $c_s = 15$ and $t = 5$, focusing on broader analysis of calculation combinations.

3.3. Model Configuration

In our experiments, we use OpenAI LLMs throughout. GPT-4o handles the planning phase for concept diversification and the review phase for response standardization and voting. For the monitoring phase's fixer component p_{θ_f} (see Fig. 2), we deploy GPT-4o-mini for cost efficiency while maintaining accuracy. The generator, evaluator, and solver components (see Fig. 2) utilize four different models (GPT-3.5-Turbo, GPT-4o-mini, GPT-4-Turbo, and GPT-4o) to enable direct comparison with other prompting methods. Therefore, GPT-3.5-Turbo+MDToC, GPT-4o-mini+MDToC, GPT-4-Turbo+MDToC, and GPT-4o+MDToC schemes each use their namesake

model for monitoring phase components, while all sharing GPT-4o for planning/reviewing and GPT-4o-mini for fixing.

To demonstrate the fairness of these comparisons despite including GPT-4o and GPT-4o-mini, we analyzed token consumption across all models and phases for three datasets in Table 1. The analysis shows that GPT-4o for the planning and reviewing phase uses less than 1% of total tokens for both CHAMP and MATH datasets and approximately 2% for Game-of-24. In the fixer component, GPT-4o-mini consumes 23,428, 21,599, and 1,945 tokens for CHAMP, MATH, and Game-of-24 respectively, accounting for 4.1% to 6.4% of total tokens in the monitoring phase. Since the backbone model is responsible for 91.6% to 95.4% of overall token usage, we can confidently make valid comparisons with alternative prompting techniques that utilize these four primary GPT models.

Table 1. Average tokens used per response of GPT-4o, GPT-3.5-Turbo, GPT-4o-mini, and GPT-4-Turbo across the planning, reviewing, and monitoring phases for the CHAMP, MATH, and Game-of-24 (G24).

GPT	Phase	Dataset		
		CHAMP	MATH	G24
4o	Planning	2,671	2,292	655
	Reviewing	346	424	127
3.5-Turbo	Generator	548,202	472,175	36,151
4o-mini		465,420	443,532	34,437
4-Turbo	Evaluator	433,588	378,745	33,804
4o		367,091	316,275	29,668
4o-mini	Fixer	23,428	21,599	1,945

Table 2. Comparative performance of different prompting approaches for various GPTs on the CHAMP dataset. '0-shot' and '5-shot' denote in-context examples in prompts; '1/3 sln' indicates the proportion of complete solution provided; CoT and CH represent Chain-Of-Thought and Annotated concepts and hints in prompts.

Prompt	GPT			
	3.5-Turbo	4o-mini	4-Turbo	4o
0-shot	28.5	36.2	41.9	55.3
CoT	29.6	36.5	37.8	55.2
5-shot	34.8	38.7	43.1	56.5
CoT + CH	34.4	42.3	53.0	60.0
1/3 sln	33.7	43.4	53.7	63.5
ToT	31.7	37.8	52.7	61.3
GoT	30.5	39.6	53.1	60.9
Our MDToC	39.5	48.3	58.1	68.2

3.4. CHAMP evaluation

Table 2 compares various prompting techniques across GPT-3.5-Turbo, GPT-4o-mini, GPT-4-Turbo, and GPT-4o models. While traditional approaches like zero-shot, CoT, 5-shot prompting show modest results, incorporating concepts and hints into prompting techniques demonstrates greater success. CoT + CH improves accuracy, with partial solution provision (1/3 sln) offering further gains. For instance, GPT-4-Turbo achieves 53.0% accuracy with CoT+CH, significantly outperforming its 37.8% CoT and 43.1% 5-shot results.

Our MDToC demonstrates superior performance across all models, achieving 39.5% (GPT-3.5-Turbo), 48.3% (GPT-4o-mini), 58.1% (GPT-4-Turbo), and 68.2% (GPT-4o). These results represent substantial improvements over ToT and GoT, with gains ranging from 5.0% to 10.5%, underscoring the effectiveness of dynamically structuring relevant concepts and

Table 3. Comparative performance of different prompting approaches for various GPTs on the MATH dataset

Prompt	GPT			
	3.5-Turbo	4o-mini	4-Turbo	4o
0-shot	45.7	71.4	72.6	81.5
CoT	48.6	72.4	73.3	81.6
5-shot	54.3	77.1	79.5	82.4
ToT	53.2	78.5	80.4	87.1
GoT	51.8	80.0	81.2	87.6
Our MDToC	60.8	83.8	86.6	89.5

employing a metacognitive feedback mechanism to refine calculation steps.

3.5. MATH evaluation

In Table 3, 0-shot prompts yield 45.7% for GPT-3.5-Turbo, 71.4% for GPT-4o-mini, 72.6% for GPT-4-Turbo, and 81.5% for GPT-4o on the MATH dataset. Adding reasoning steps through CoT improves these scores slightly (e.g., from 45.7% to 48.6% for GPT-3.5-Turbo), while increasing the number of examples (5-shot) provides further gains (up to 79.5% for GPT-4-Turbo and 82.4% for GPT-4o). The ToT and GoT methods surpass standard few-shot prompts for the two more advanced models, pushing GPT-4-Turbo to 80.4% and 81.2% and GPT-4o to 87.1% and 87.6%, a notable jump of around 7% from CoT.

Even so, our MDToC outperforms all these strategies, achieving 60.8% for GPT-3.5-Turbo, 83.8% for GPT-4o-mini, 86.6% for GPT-4-Turbo, and 89.5% for GPT-4o. Compared to ToT, our MDToC provides an extra 7.6% boost for GPT-3.5-Turbo, 5.3% for GPT-4o-mini, and 6.2% for GPT-4-Turbo. These results strengthen our claim that our MDToC not only overcomes the evaluation constraints in ToT but also achieves more robust performance than purely tree-based approaches like ToT.

3.6. Game-of-24 evaluation

Table 4 reports six prompting strategies (0-shot, CoT, 5-shot, ToT, GoT, and our MDToC) evaluated on four models (GPT-3.5-Turbo, GPT-4o-mini, GPT-4-Turbo, and the GPT-4o). The broad pattern remains: minimal prompting (0-shot or CoT) yields very low accuracy (2-10%), while adding a handful of demonstrations (5-shot) produces a modest gain (6-18%). ToT then brings a significant jump for GPT-4o-mini to 56%, GPT-4-Turbo to 74%, and GPT-4o to 88%. GoT raises scores further to 62%, 81%, and 90% for GPT-4o-mini, GPT-4-Turbo,

Table 4. Comparative performance of different prompting approaches for various GPTs on the Game-of-24 dataset.

Prompt	GPT			
	3.5-Turbo	4o-mini	4-Turbo	4o
0-shot	2	3	4	10
CoT	3	3	4	9
5-shot	6	8	10	18
ToT	19	56	74	88
GoT	21	62	81	90
Our MDToC	30	75	85	90

Table 5. Per-case cost and accuracy for our MDToC with GPT models on MATH, CHAMP, and Game-of-24 (G24). W stands for using GPT-4o-mini in the fixer component and GPT-4o in the planning and reviewing phase in our MDToC configuration. W/o stands for using the same LLMs across all components.

Dataset	GPT	Cost (\$)		Accuracy	
		w	w/o	w	67
MATH	3.5-Turbo	0.58	0.62	60.8	54.1
	4-Turbo	19.79	21.16	86.6	86.9
	4o-mini	0.23	0.23	83.8	82.9
	4o	3.5	3.7	89.5	89.7
CHAMP	3.5-Turbo	0.69	0.72	39.5	33.9
	4-Turbo	22.59	24.15	58.1	59.0
	4o-mini	0.24	0.24	48.3	47.1
	4o	3.94	4.08	68.2	68.5
G24	3.5-Turbo	0.03	0.05	30	25
	4-Turbo	1.79	1.92	85	85
	4o-mini	0.02	0.02	75	74
	4o	0.37	0.39	90	90

and GPT-4o, respectively; even GPT-3.5-Turbo climbs from 19% (ToT) to 21% (GoT).

Despite these gains, our MDToC still delivers the best accuracy across the board: 30% (+9% over GoT) on GPT-3.5-Turbo, 75% (+13%) on GPT-4o-mini, 85% (+4%) on GPT-4-Turbo, and a top-tied 90% on GPT-4o. These margins underscore the critical role of MDToC’s evaluator p_{θ_e} and fixer p_{θ_f} , which detect and repair erroneous intermediate expressions involving the four numbers in Game-of-24, maintaining logical consistency and reducing hallucinations throughout the reasoning chain.

3.7. Cost analysis of our MDToC configuration

Replacing the planner and reviewer with GPT-4o and the fixer with GPT-4o-mini reduces or maintains costs while maintaining or improving accuracy across all backbone models and datasets tested (shown in

Table 6. Hyperparameters (see Sections 2.4.1 and 2.4.2) in MDToC with GPT-4o on MATH and Game-of-24 (G24). Acc stands for Accuracy.

$(C_{min}, C_{max}, SC_{min}, SC_{max})$	(c_s, t)	MATH Acc	G24 Acc	MATH Cost	G24 Cost
(2,4,1,2)	(2,10)	89.5%	87%	\$3.5	\$0.7
(3,5,1,2)	(2,10)	89.6%	88%	\$4.9	\$0.8
(3,5,2,4)	(3,15)	89.9%	88%	\$5.8	\$1.0
(1,1,1,1)	(15,5)	83.4%	90%	\$2.6	\$0.4

Table 5). For GPT-3.5-Turbo, costs decreased while accuracy increased on MATH (54.1% to 60.8%), CHAMP (33.9% to 39.5%), and Game-of-24 (25% to 30%). GPT-4-Turbo saw cost reductions with minimal accuracy changes on all datasets, while GPT-4o as backbone showed 3-5% cost savings with negligible accuracy differences ($\leq 0.3\%$). These results demonstrate that delegating auxiliary roles to lighter models is an effective strategy for reducing computational expenses without compromising performance, with weaker models particularly benefiting from the diverse planning concepts provided by GPT-4o.

3.8. Hyperparameter Sensitivity Test

Table 6 shows important hyperparameter combinations. Increasing concept exploration to 5 concepts ($C_{max} = 5$) and up to 20 sub-concepts in total ($SC_{max} = 4$) only marginally improves the accuracy of our MDToC on MATH (by 0.4%). This confirms that about 4 concepts ($C_{max} = 4$) are sufficient. In addition, the cost is significantly lower (\$3.5 versus \$5.8). For Game-of-24, accuracy slightly decreases (from 90% to 88%) when computational exploration is narrowed in favor of concept variety, indicating that this dataset does not benefit from diverse concepts. Meanwhile, despite the cost decrease (down to \$2.6), limiting concepts while broadening computational exploration reduces MATH accuracy (to 83.4%). However, such settings suit exploration-based datasets like Game-of-24.

3.9. Hyperparameter Selection Heuristics

Table 6 suggests a three-step procedure for selecting the parameters of the dynamic tree on a new dataset without exhaustive trial-and-error.

Step 1: Determine whether the dataset benefits from concept exploration or

computational exploration. To do so, inspect the annotated concepts and solutions of a small number of problems. Datasets whose problems require different mathematical concepts (CHAMP, MATH) benefit from concept exploration. In contrast, exploration-based datasets whose problems require diverse calculation combinations (Game-of-24) benefit from computational exploration. This choice has the largest effect in Table 6: removing concept exploration lowers MATH accuracy by 6.1% (89.5% to 83.4%), whereas it raises Game-of-24 accuracy by 3% (87% to 90%).

Step 2: Set the parameters according to Step

1. For concept exploration, use the concept range, sub-concept range, c_s , and t of Section 3.2. Wider ranges are not worthwhile: Table 6 shows that they improve MATH accuracy by at most 0.4% at 40% to 66% higher cost. For computational exploration, set all four tree parameters to 1. Then, use a larger c_s with fewer iterations t (Section 3.2). This setting gives the highest Game-of-24 accuracy (90%) at the lowest cost (\$0.4) in Table 6.

Step 3: Confirm the parameters on a small development set of, for example, 20 to 30 held-out problems. Increase C_{max} and SC_{max} by one at a time. Stop when the accuracy gain per additional cost falls below a preset threshold, for example 1% accuracy per additional dollar of per-case cost. In Table 6, the first increment (from $C_{max} = 4$ to 5) gains only 0.1% for \$1.4 more per case. This is far below such a threshold. Therefore, the confirmation stops after one increment.

4. Conclusions

Our proposed MDToC framework enhances mathematical reasoning in LLMs through the metacognitive stages of planning, monitoring, and reviewing. It outperforms ToT and GoT techniques, achieving up to 19% higher accuracy than ToT on Game-of-24 and showing consistent improvements on CHAMP and MATH. Our MDToC excels in calculation-intensive tasks through dynamic concept structuring and iterative error correction, establishing a foundation for future research in complex problem-solving.

5. Study limitations

Despite MDToC's superior performance compared to previous prompting methodologies, several notable limitations affect its practical implementation. First, our metacognitive calculation approach exhibits domain-specific constraints, particularly

in mathematical fields such as geometry, where spatial reasoning predominates over calculation verification. In such domains, the iterative verification processes central to MDToC may offer diminishing returns, as the primary cognitive challenges relate to geometry visualization rather than computational validation. Two mitigations could extend MDToC to such domains. First, the generator and evaluator components could use multimodal LLMs, so that each step is evaluated against the figure itself. Second, an explicit spatial parsing tool could translate geometric figures into symbolic representations such as coordinates and angle relations, as in AlphaGeometry [32]. These representations would make spatial reasoning steps evaluable in the same way as calculations. We leave this to future work.

Second, the performance improvements delivered by our MDToC incur computational costs. As demonstrated in our analyses (see Tables 1 and 5), the method introduces significant resource overhead, particularly when implementing MDToC with expensive LLMs such as GPT-4-Turbo. Token consumption for CHAMP and MATH benchmark problems reaches approximately 450,000 tokens per problem, translating to approximately \$20 per problem, a cost scale that may prove prohibitive for educational institutions and research organizations with limited budgets. These economic constraints potentially restrict MDToC's deployment in resource-limited environments.

Third, the best parameters of the dynamic tree differ between datasets that benefit from concept exploration and exploration-based datasets (Table 6). Section 3.9 provides a selection procedure, but a small development set is still needed to confirm the parameters on a new dataset. In addition, the dynamic tree of concepts requires the backbone LLM to follow the dependencies among concepts, sub-concepts, and calculations. Such deep relational reasoning remains a limitation even for strong LLMs such as DeepSeek-R1 [33].

CRediT author statement

- **Tung Duong Ta:** Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing original draft, Visualization.
- **Tim Oates:** Writing - review and editing.
- **Thien Van Luong:** Writing - review and editing.
- **Huan Vu:** Writing - review and editing.
- **Tien Cuong Nguyen:** Writing - review and editing.

COMPETING INTERESTS

The authors declare that there is no conflict of interest regarding the publication of this article.

REFERENCES

- [1] J. Achiam, S. Adler, S. Agarwal, et al. (2023), "GPT-4 technical report", *arXiv*, DOI: 10.48550/arXiv.2303.08774.
- [2] Anthropic (2024), "The Claude 3 model family: Opus, Sonnet, Haiku", *Claude-3 Model Card*, **1**, pp.1-42.
- [3] K. Cobbe, V. Kosaraju, M. Bavarian, et al. (2021), "Training verifiers to solve math word problems", *arXiv*, DOI: 10.48550/arXiv.2110.14168.
- [4] A. Patel, S. Bhattamishra, N. Goyal (2021), "Are NLP models really able to solve simple math word problems?", *arXiv*, DOI: 10.48550/arXiv.2103.07191.
- [5] D. Hendrycks, C. Burns, S. Kadavath, et al. (2021), "Measuring mathematical problem solving with the MATH dataset", *arXiv*, DOI: 10.48550/arXiv.2103.03874.
- [6] Y. Mao, Y. Kim, Y. Zhou (2024), "CHAMP: A Competition-level Dataset for Fine-Grained Analyses of LLMs' Mathematical Reasoning Capabilities", *arXiv*, DOI: 10.48550/arXiv.2401.06961.
- [7] N. Patel, M. Kulkarni, M. Parmar, et al. (2024), "Multi-LogiEval: Towards Evaluating Multi-Step Logical Reasoning Ability of Large Language Models", *arXiv*, DOI: 10.48550/arXiv.2406.17169.
- [8] N. Tyagi, M. Parmar, M. Kulkarni, et al. (2024), "Step-by-Step Reasoning to Solve Grid Puzzles: Where do LLMs Falter?", *arXiv*, DOI: 10.48550/arXiv.2407.14790.
- [9] P.Y. Wang, T.S. Liu, C. Wang, et al. (2026), "A survey on large language models for mathematical reasoning", *ACM Computing Surveys*, **58(8)**, pp.1-35.
- [10] J. Wei, X. Wang, D. Schuurmans, et al. (2022), "Chain-of-thought prompting elicits reasoning in large language models", *Advances in Neural Information Processing systems*, **35**, pp.24824-24837.
- [11] X. Wang, J. Wei, D. Schuurmans, et al. (2023), "Self-consistency improves chain of thought reasoning in language models", *arXiv*, DOI: 10.48550/arXiv.2203.11171.
- [12] S. Yao, D. Yu, J. Zhao, et al. (2024), "Tree of thoughts: Deliberate problem solving with large language models", *Advances in Neural Information Processing Systems*, **36**, DOI: 10.48550/arXiv.2305.10601.
- [13] J. Long (2023), "Large language model guided tree-of-thought", *arXiv*, DOI: 10.48550/arXiv.2305.08291.
- [14] M. Besta, N. Blach, A. Kubicek, et al. (2024), "Graph of thoughts: Solving elaborate problems with large language models", *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, pp.17682-17690.
- [15] Y. Yao, Z. Li, H. Zhao (2023), "Beyond chain-of-thought, effective graph-of-thought reasoning in language models", *arXiv*, DOI: 10.48550/arXiv.2305.16582.
- [16] O. Fagbohun, R.M. Harrison, A. Dereventsov (2024), "An empirical categorization of prompting techniques for large language models: A Practitioner's Guide", *arXiv*, DOI: 10.48550/arXiv.2402.14837.
- [17] E.R. Lai (2011), "Metacognition: A literature review", *Always learning: Pearson Research Report*, **24**, pp.1-40.
- [18] Y. Wang, Y. Zhao (2023), "Metacognitive prompting improves understanding in large language models", *arXiv*, DOI: 10.48550/arXiv.2308.05342.
- [19] T. Kojima, S.S. Gu, M. Reid, et al. (2022), "Large language models are zero-shot reasoners", *Advances in Neural Information Processing Systems*, **35**, pp.22199-22213.
- [20] T. Brown, B. Mann, N. Ryder, et al. (2020), "Language models are few-shot learners", *Advances in Neural Information Processing Systems*, **33**, pp.1877-1901.
- [21] Y. Zhou, Z. Liu, J. Jin, et al. (2024), "Metacognitive retrieval-augmented large language models", *Proceedings of the ACM on Web Conference 2024*, pp.1453-1463.
- [22] A. Didolkar, A. Goyal, N.R. Ke, et al. (2024), "Metacognitive capabilities of LLMs: An exploration in mathematical problem Solving", *arXiv*, DOI: 10.48550/arXiv.2405.12205.
- [23] X. Chen, R. Aksitov, U. Alon, et al. (2023), "Universal self-consistency for large language model generation", *arXiv*, DOI: 10.48550/arXiv.2311.17311.
- [24] A. Madaan, N. Tandon, P. Gupta, et al. (2023), "Self-refine: Iterative refinement with self-feedback", *Advances in neural Information Processing Systems*, **36**, pp.46534-46594.
- [25] Y. Li (2025), "Decomposing LLM self-correction: The accuracy-correction paradox and error depth hypothesis", *arXiv*, DOI: 10.48550/arXiv.2601.00828.
- [26] Q. Long, K.J. Jiang, J. Chen, et al. (2026), "Self-verification dilemma: experience-driven suppression of Overused Checking in LLM reasoning", *arXiv*, DOI: 10.48550/arXiv.2602.03485.
- [27] Z. Shao, P. Wang, Q. Zhu, et al. (2024), "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models", *arXiv*, DOI: 10.48550/arXiv.2402.03300.
- [28] J. Zhang, G. Ma, S. Liu, et al. (2026), "STRIDE: Learnable stepwise language feedback for LLM Reasoning", *arXiv*, DOI: 10.48550/arXiv.2605.18851.
- [29] K.Y.L. Ku, I.T. Ho (2010), "Metacognitive strategies that enhance critical thinking", *Metacognition and Learning*, **5(3)**, pp.251-267.
- [30] AoPS (2025), "AIME Problems", *Art of Problem Solving*, https://artofproblemsolving.com/community/c3416_aime_problems, accessed 10 January 2025.
- [31] AoPS (2025), "AMC 10 Math Problems", *Art of Problem Solving*, https://artofproblemsolving.com/community/c3414_amc_10, accessed 10 January 2025.
- [32] T.H. Trinh, Y. Wu, Q.V. Le, et al. (2024), "Solving olympiad geometry without human demonstrations", *Nature*, **625(7995)**, pp.476-482.
- [33] C.C. So, Y. Sun, J.M. Wang, et al. (2025), "Are large language models capable of deep relational reasoning? insights from DeepSeek-R1 and benchmark comparisons", *2025 IEEE International Conference on Artificial Intelligence Testing (AITest)*, pp.168-177.

Journal on Information Technologies & Communications: Manuscript Template

General Instructions on using this template and submitting a manuscript to Journal on Information Technologies & Communications: Thank you for preparing a manuscript for submission to Journal on Information Technologies & Communications. Using this template, or following the guidelines below, will help us in processing your article.

The structure of different types of articles may vary according to the type of study, and the content of the article.

The following are the detailed main components of the original research and review articles submitted to the journal.

From 2025 onwards, you can submit your article at: <https://mc04.manuscriptcentral.com/mostvjstc>

Original research

The original research should have three following parts:

Preliminary section: It includes title, list of all contributing authors with their institutional affiliation, and abstract (160-250 words) with at least 3 keywords.

Title: How to format a science article (replace with your real title)

Author(s): List the author name(s) separated by commas. Use superscript numbers to link affiliations, and symbols *†‡ for author notes. For example, Y. Obsie¹, S.A. Adem^{1,2}

***Corresponding author(s):** Include the email addresses or telephone numbers of the corresponding author(s). Please use the asterisk (*) symbol for the corresponding author information. This information is at the bottom of the manuscript's first page.

Author's affiliation addresses:

¹*Affiliations should be preceded by superscript numbers corresponding to the author list*

²*For any institutions, please provide the institution name, street, district, city, country*

Received ; revised ; accepted

Abstract:

Start your abstract here...

Keywords: keyword 1, keyword 2, keyword 3...

Abstract: The abstract should range between 160 and 250 words and be structured in a single paragraph following keywords in a separate line. The abstract should represent the main part of the research manuscript. It should reflect a brief and comprehensive background following the aim of the study (main questions to be addressed), methods used for investigating the problem, a summary of significant findings, main conclusions, and implications of the results.

Keywords: They should come immediately after the abstract and contain 3-7 specific and relevant keywords. They should reflect the research study and the relevant field of research.

Classification numbers: Depend on the Journal's Classification Scheme.

Author(s) are encouraged to write a "Highlights" that summarises the most exciting, interesting points from your manuscript.

Manuscript section: It is the main body of the scientific manuscript including: 1) Introduction; 2) Material(s) and method(s); 3) Results and discussion (separation of these two sections is acceptable); 4) Conclusions; 5) Recommendations/Suggestions/Implications/Study limitations...(if any). These sections may slightly vary depending on the type of research approach and design used in the research. Each section should be numbered.

All of the Figures, Tables, and Equations should be cited in numerical order.

Figures

Fig. #: (Begin each figure caption with a label, "Fig. 1." for example, as a new paragraph).

Fig. 1. The figure caption should begin with an overall descriptive statement of the figure followed by additional text. They should be immediately after each figure. Figure parts are indicated with capital letters (A). If you prefer, you can place both the actual figures and captions logically through the text near where they are cited rather than at the end of the file (but not both). If a paragraph in the main text begins with the name of

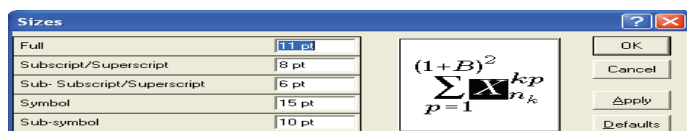


Fig. 1. The figure caption.

a figure, write out "Figure" in full (e.g., <para> "Figure 1 shows...").

Tables

Table #: (Begin each table caption with a label "Table 1.", etc. as a new paragraph).

Table 1. Start this caption with a short description of your table. Format tables

using the Word Table commands and structures. Do not create tables using spaces or tabs characters.

Table 1. Caption.

a	b	c	d	e
f				
g				
h				

Equations

$$\int_a^b x dx \quad (1)$$

Equations can be included. We do not recommend using the native Word 2007, 2008, 2010 or 2011 equation editor. This can in some cases produce less reliable MathML, the online markup language we use, which may result in display errors. Instead, use the legacy equation editor in Word (Insert menu; select insert object; select word equation) or use MathType (recommended). If you enter equations in simple LaTeX, check that they will convert accurately (Word 2007 and higher can convert simple LaTeX equations).

Closing section: It includes appendices/supplementary materials/data availability statement (if applicable), CRediT author statement (for multiple authors), acknowledgements (if applicable), competing interests, ethics (if any) and a complete list of references corresponding to all the citations given in the text of the manuscript. The appendices/supplementary materials may include data files and other materials relevant to the research that should be submitted in separate files if required.

APPENDICES (if any)

The appendices should ideally be under 1000 words (approximately 2 A4 pages) for printing version. With longer appendices, they are only accepted for online publication.

CRediT author statement (for multiple authors)

CRediT offers authors the opportunity to share an accurate and detailed description of their diverse contributions to the published work.

ACKNOWLEDGEMENTS (if applicable)

This section is for the researchers who were funded for the research projects, and it should enclose all sources of financial support.

COMPETING INTERESTS

The author(s) declare(s) that there is no conflict of interest regarding the publication of this article.

REFERENCES

For in-text citation: References should be cited in the bracket with a number [1]. Multiple reference citations are separated by commas [2, 3] or if a series, dashes [4-6]. When citing sources with the author(s) and the year, they should be shortened to the first author's name, followed by et al. (with more than one author) and the year.

For reference list: With less than or equal to three authors, list all the author(s). With more than three authors, the first three should be listed and followed by et al.

References are listed in numerical order, and in the same order in which they are cited in text. The reference list should include all and only those references you have cited in the text. The reference list appears at the end of the article.

The reference consists of: Initial(s) and first/last name of the author(s), author's names separated by commas (Year of publication), "Titles of journal articles or chapter of book", *Title of Journal, Book or Other Publication*, **Volume(Number)** (if any), page numbers (if any), DOI (if any), URL and accessed date (if any) (in its original language if not English).

To cite a website and online/electronic resources, please include the accessed date (day month year).

Example of a reference list:

[1] T. Binzoni, T.S. Leung, D. Boggett, et al. (2003), "Non-invasive laser Doppler perfusion measurements of large tissue volumes and human skeletal muscle blood RMS velocity", *Phys. Med. Biol.*, **48(1)**, pp.2527-2549, DOI: 10.1088/0031-9155-48-15-318.

[2] H.W. Bischoff, H.C. Bold (1963), *Some Soil Algae from Enchanted Rock and Related Algal Species*, University of Texas Publications, 95pp.

[3] R. Busi (2016), "Weedy rice in the Philippines and Vietnam", <https://www.aciar.gov.au/publication/technical-publications/weedy-rice-philippines-and-vietnam-final-report>, accessed 15 July 2018.

Review articles

They should be different from the original research as they would contain different sections on the literature review. Authors may structure this part as per nature, background, and available literature on the issue argued in the manuscript.

For more detailed instructions, please visit our website: ictmag.ictvietnam.vn/ict

EDITORIAL BOARD

Editorial Board

Chairman

Le Hoai Bac

University of Science, Vietnam National University, Ho Chi Minh City, Vietnam

Members

Giuseppe D'Aniello

University of Salerno, Italy

Tzung-Pei Hong

National University of Kaohsiung, Taiwan, China

Nguyen Khanh Van

Hanoi University of Science and Technology, Vietnam

Nguyen Thanh Thuy

University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

Nguyen Xuan Huan

Middlesex University London, United Kingdom

Nguyen Xuan Huy

Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam

Tran Xuan Nam

Le Quy Don Technical University, Vietnam

Tu Minh Phuong

Posts and Telecommunications Institute of Technology, Vietnam

Editor-in-Chief

Nguyen Thi Huong Giang

Ministry of Science and Technology, Vietnam

Deputy Editor-in-Chief

Tran Anh Tu

Ministry of Science and Technology, Vietnam

Pham Thi Minh Nguyet

Ministry of Science and Technology, Vietnam

Managing Editors

Phi Cong Thuong

Ministry of Science and Technology, Vietnam

Le Thi Bac

Ministry of Science and Technology, Vietnam

Nguyen Lan Phuong

Ministry of Science and Technology, Vietnam

Tang Xuan Binh

Ministry of Science and Technology, Vietnam

Truong Thi Thao Duong

Ministry of Science and Technology, Vietnam

Phung Thi Tuyet Mai

Ministry of Science and Technology, Vietnam

EDITORIAL OFFICE

115 Tran Duy Hung Street, Yen Hoa Ward, Hanoi, Vietnam

Tel: (84.24) 39436793; Fax: (84.24) 39436794

Email: cscntt@mst.gov.vn

Website: <https://ictmag.ictvietnam.vn/ict>

PUBLICATION LICENCE

No. 459/GP-BTTTT 20th July 2021

No. 50/GP-BTTTT 1st February 2023

No. 318/GP-BTTTT 29th October 2024

No. 39/GP-CBC 1st November 2024