Robust Vietnamese Date-Arithmetic Question Answering through QDoRA Fine-Tuning and Canonicalized Decoding

Do Tien Dung and **Vu Minh Dang** and **Ho Tu Minh** and **Le Tuan Anh** and **Do Quang Dung**
University of Engineering and Technology, Vietnam National University

Abstract

Temporal reasoning is one of the fundamental properties of natural language comprehension, and date arithmetic is one of the important problems in the symbolic processing of time. This paper describes our approach and outcome for the Date Arithmetic sub-task of the VLSP 2025 Temporal QA Challenge. The task asks systems to take Vietnamese questions as input, recognize temporal referential and operations, and compute a new date accordingly from a given context. To counter this, we took a fine-tuning strategy on Vistral, which is a 7-billion parameter language model. A key point in our strategy was the use of data augmentation techniques, whereby we could develop a stronger and more generalized model that would comfortably be able to handle the different question styles from the TimeBench dataset. Our system worked extremely well, achieving 99% accuracy on the private test set and ranking first among all the participants. In this work, we give a clear explanation of our approach, an interpretation of the results, and discuss what we learned in fine-tuning large models to individual reasoning tasks, highlighting the strength of data augmentation to achieve almost perfect performance.

Source code available at:
https://github.com/minhlake04/VSLP-2025-temporal-qa-date-arithmetic-submission

1 Introduction

Temporal reasoning is one of the foundations of cutting-edge natural language understanding, enabling machines to comprehend and reason over the ordering, duration, and sequence of events described in text. This capability is essential for a wide range of applications, from question answering and information extraction to dialogue systems. Date arithmetic is one among the numerous temporal reasoning challenges that is a simple symbolic manipulation problem. It must be extremely accurate in terms of time and also capable of making precise calculations, as a basis for further temporal logic which will be more complex.

We introduce our research in the setting of the Date Arithmetic (date-arith) sub-task of the Vietnamese Language and Speech Processing (VLSP) 2025 Temporal QA Challenge. The shared task presents a wonderful opportunity to benchmark and develop temporal reasoning systems for Vietnamese. The task can be described as follows:

- **Input**: A question in Vietnamese, which contains a base date, D_{base}, and a temporal offset operation, O_{offset}, indicating a duration to be added or subtracted.
- **Output**: The target date, D_{target}, calculated by applying the operation to the base date: $D_{target} = D_{base} \pm O_{offset}$.

The test data for this task is a Vietnamese translation and extension of the Date Arithmetic part of the complete TimeBench benchmark, ensuring a challenging and diverse set of questions.

In this paper, we present our methodology and outcome of our system which was ranked number one in the competition. Our approach revolves around fine-tuning Vistral, a powerful 7-billion-parameter language model, to produce the computed date directly from the input question. One key element of our approach was the heavy reliance on data augmentation strategies, enabling us to develop a stronger model that could accommodate a vast number of linguistic forms for calculating dates. Our end system attained 99% accuracy on the private test set, proving the benefit of combining a strong base model and purposeful data improvement.

The rest of this paper is organized as follows: Section 2 provides an overview of related works on temporal reasoning. Section 3 describes our proposed approach, i.e., the model structure and the data augmentation process. Section 4 gives a complete analysis of our experimental results. Finally, Section 5 discusses our findings and concludes with some possible directions for future work.

2 Related Work

2.1 Temporal Reasoning Benchmarks in NLP

Evaluating the temporal reasoning of Large Language Models (LLMs) has been the focus of several key benchmarks. For instance, datasets like **TORQUE** (Ning et al., 2020) and **MC-TACO** (Zhou et al., 2019) have consistently revealed significant gaps between LLM and human performance on complex tasks involving event ordering and temporal commonsense. More comprehensively, the **TimeBench** benchmark (Chu et al., 2024) demonstrated that even state-of-the-art LLMs struggle with fundamental symbolic manipulations like date arithmetic, highlighting the need for targeted, domain-specific solutions.

2.2 Temporal Reasoning in Vietnamese NLP

Research in Vietnamese temporal reasoning is a nascent field. Early efforts primarily focused on the foundational tasks of time expression recognition and normalization (Lambert and Nguyen, 2012; Strötgen and Nguyen, 2014), rather than complex question answering. A significant step forward was the **VLSP 2025 Temporal QA challenge**, which introduced the first dedicated benchmark for temporal reasoning in Vietnamese by adapting tasks from TimeBench (Association for Vietnamese Language and Speech Processing, 2025). Our work directly addresses the symbolic Date Arithmetic sub-task from this challenge.

2.3 Parameter-Efficient Fine-Tuning of LLMs

To efficiently adapt LLMs for specialized tasks, we employ **Parameter-Efficient Fine-Tuning (PEFT)** . This field has progressed rapidly from **Low-Rank Adaptation (LoRA)**, which injects small, trainable low-rank matrices into a frozen model (Hu et al., 2021), to **QLoRA**, which integrates 4-bit quantization to dramatically reduce memory requirements (Dettmers et al., 2023). A recent innovation, **Weight-Decomposed LoRA (DoRA)**, further improves performance by decomposing weights into magnitude and direction components, allowing for more expressive and stable training updates (Liu et al., 2024). Our methodology builds on this paradigm, leveraging the efficiency and power of these techniques to specialize an LLM for date arithmetic without the prohibitive cost of full fine-tuning.

3 Methodology

3.1 Overview of Method: Fine-tuning Vistral-7B-iSMART with QDoRA and EM-oriented Decoding

3.1.1 Data Preparation, Augmentation, and Formatting

We began with a Vietnamese date-arithmetic corpus derived from TimeBench via translation and rule-based templating, and expanded it with a retrieval-augmented generation (RAG) pipeline coupled with a deterministic solver filter. Concretely, we index $\sim$2k seed questions together with five calendar rules (leap years, month lengths, month-rollover, week=7 days, output canonicalization) using a TF-IDF retriever (min_df=2, max_features=30k). For each of 300 randomly sampled seeds (random.seed=42), we retrieve top 5 snippets, prompt Google Gemini to propose up to 8 paraphrases per seed, and de-duplicate against the seed set (case-insensitive). Each candidate question must contain a recognizable temporal anchor and is parsed by a strict regex-based recognizer into one of three granularities - day (dd/mm/yyyy), month ("Tháng m, yyyy"), or year - together with an offset signature covering years/months/weeks/days and a direction token ("trước/-" vs. "sau/+"). Candidates are accepted only if a rule-consistent solver $(\text{dateutil.relativedelta})$ returns a canonical answer string; otherwise they are discarded. This process yields 1,949 additional, solver-validated items, bringing the final training set to 4,949 question - answer pairs, all wrapped in a fixed instruction schema $(\text{\#\#\# Instruction: \{question\} \#\#\# Response: \{answer\}.})$ to stabilize loss masking and match inference prompts.

Beyond pipeline mechanics, we characterize the synthesized 1,949-item subset along three axes that are enforced by construction in the code:

1. **Question taxonomy:** questions are stratified by anchor granularity (day/month/year) detected via dedicated regexes; lexical variants include both digit forms and Vietnamese number words (e.g., "một," "hai," "mười hai"), ensuring coverage of common surface realiza-

tions while preserving parseability.

2. **Operation typology:** offsets span single-unit (years, months, weeks, days) and mixed-unit compositions; directionality is normalized to addition/subtraction via a compact sign detector keyed to "trước/bớt/trừ" (-) and "sau/thêm/+" (+). End-of-month and leap-year corner cases are explicitly exercised and resolved by a clamp-to-valid-date policy, guaranteeing valid Gregorian outcomes.

3. **Quality and validity:** syntactic validity is guaranteed by anchor detection; semantic validity is guaranteed by the solver, which encodes the same calendar rules used at retrieval time. Canonical answer formats are enforced ("dd/mm/yyyy" for day-level, "Tháng m, yyyy" for month-level, and bare "yyyy" for year-level), yielding deterministic supervision targets.

Taken together, the augmentation improves linguistic diversity (via retrieval-conditioned paraphrasing) without sacrificing correctness (via solver filtering) and produces a balanced portfolio of anchor granularities and offset structures suitable for EM-oriented training. We note two limitations intrinsic to the current code: (i) the anchor recognizer prioritizes numeric date forms and a limited lexicon of Vietnamese number words, potentially under-sampling very long-form dates; and (ii) week-level offsets are ignored for month/year anchors by design to preserve granularity. These constraints can be relaxed in future iterations by expanding the lexical map, adding additional date patterns, and introducing unit-conversion rules for cross-granularity reasoning.

3.1.2 Model Fine-tuning with QDoRA (DoRA on QLoRA)

We fine-tuned Vistral-7B-iSMART, a Vietnamese Mistral-class 7B model, using a QDoRA setup: DoRA adapters on top of a 4-bit (NF4) quantized base (QLoRA). Concretely, the 4-bit quantization (bitsandbytes) reduces GPU memory, while DoRA constrains adaptation to the direction of the weights (magnitude held fixed), improving optimization stability and preserving pre-training knowledge. We targeted attention and MLP projections (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj) with rank $R = 128$ and $\alpha = 256$ and applied a modest LoRA dropout of 0.05.

To prevent the model from learning instruction boilerplate, we used an answer-only loss mask: tokens before the sentinel ### Response: are set to -100 in the labels, so only the completion segment contributes to the loss. This sharply focuses learning on mapping questions to canonical dates.

Optimization & hardware. Training ran on 2× RTX 5090 GPUs with bfloat16 compute when available, gradient checkpointing, AdamW (torch-fused), a cosine schedule with 10% warmup, learning rate of 10^{-4}, for 3 epochs, and an effective batch size of 16 (per-device batch of 2×8 gradient accumulation steps). This configuration achieved strong generalization while keeping memory and wall-time modest.

Base model. Vistral-7B-iSMART is an Apache-2.0 licensed, Vietnamese-oriented Mistral-7B derivative (7.29B params) fine-tuned from Viet-Mistral/Vistral-7B-Chat; the model card notes prior SFT via TRL/Unsloth. We use it strictly as the initialization checkpoint for our task-specific fine-tuning.

3.1.3 Decoding and EM-oriented Post-processing

Because evaluation uses Exact Match on canonical date strings, we adopted a decoding/post-processing stack explicitly tuned for EM:

- **Greedy decoding (no sampling):** We use do_sample=False and temperature=0.0, with an instruction-aligned prompt identical to the one used in training.

- **Canonicalization by regex.** Model outputs are normalized to exactly one of two formats:
  - Tháng m, yyyy for month-granularity anchors, or
  - dd/mm/yyyy for day-level anchors (zero-padded dd/mm).

  Any tokens after a spurious "### ..." marker are stripped.

- **Fallback solver.** If the string does not match either regex (which is rare), the same deterministic solver used during augmentation is executed on the input question to compute the date and re-format it to the canonical form.

This pipeline eliminates formatting noise and recovers otherwise-correct predictions, yielding a "free" EM boost without accessing any ground-truth answers.

3.2 Setup

3.2.1 Overview

Objective. Our objective is to fine-tune a 7-billion parameter language model for Vietnamese to solve date-arithmetic questions and return answers in a canonical format, optimized for Exact Match (EM) evaluation.

Data Preparation & Augmentation.

- We start with the TimeBench dataset (approximately 2,500 seed samples), translated into Vietnamese.
- The dataset is expanded by combining the TF-IDF retrieval method with Gemini-based paraphrasing techniques.
- All synthetic data is filtered through a deterministic solver (dateutil.relativedelta) to ensure logical correctness.
- The final dataset contains 4,949 question-answer pairs, formatted with the template: "### Instruction: {question} ... ### Response: {answer}".

QDoRA Architecture.

- **Base Model:** Vistral-7B-iSMART (7.29 billion parameters).
- **Quantization:** Utilizes QLoRA with 4-bit NF4 quantization to save memory.
- **Adapters:** DoRA adapters are applied to the attention and MLP projection layers (q/k/v/o and gate/up/down) with a rank of $R = 128$ and $\alpha = 256$.

Training Configuration.

- Utilizes answer-only loss masking (labels are set to -100 before the "### Response:\n" marker).
- Training is conducted for 3 epochs using the AdamW optimizer with a cosine schedule, a learning rate of 10^{-4}, 10% warmup, and an effective batch size of 16.
- The process uses 2 RTX 5090 GPUs, applying gradient checkpointing and bf16 precision where available.

Inference Pipeline.

- **Decoding:** Uses greedy decoding with temperature=0 and do_sample=False.
- **Canonicalization:** A regex-based normalizer converts the output to one of two formats: "Tháng m, yyyy" or "dd/mm/yyyy".
- **Fallback Solver:** A deterministic solver handles edge cases (e.g., "trước/sau", leap years, end-of-month) if the output does not match the regex.

Output & Evaluation.

- The final output is a canonical date string, ready for EM evaluation.
- EM accuracy is improved by the combination of strict normalization and a deterministic fallback, with all code paths designed to be reproducible.

Why This Approach is Effective (Key Features). The system's success stems from the synergy of its components. 4-bit quantization reduces memory requirements, DoRA's direction-only adaptation stabilizes the fine-tuning process and preserves the base model's reasoning capabilities, and the EM-oriented post-processing pipeline recovers logically correct predictions, converting them into high-accuracy, canonically formatted results.

3.2.2 Training Pipeline (Algorithm)

Objective. We use a standard causal-LM loss, masked so that only tokens after the ### Response: sentinel contribute to the loss. Let the input tokens be $x_{1:T}$ and let t_0 be the index of the first token after the sentinel. The loss is:

$$\mathcal{L} = -\sum_{t=t_0}^{T} \log p_\theta(x_t \mid x_{<t}). \tag{1}$$

This removes the gradient from the instruction boilerplate and directly optimizes the output string that Exact Match (EM) checks.

Adapters (QDoRA). We apply DoRA to the linear projections in the attention and MLP layers (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj). Conceptually, DoRA decomposes each base weight into a magnitude and a direction; the base magnitude is kept fixed, while the direction is adapted via low-rank updates. Under 4-bit QLoRA storage of the base model, this

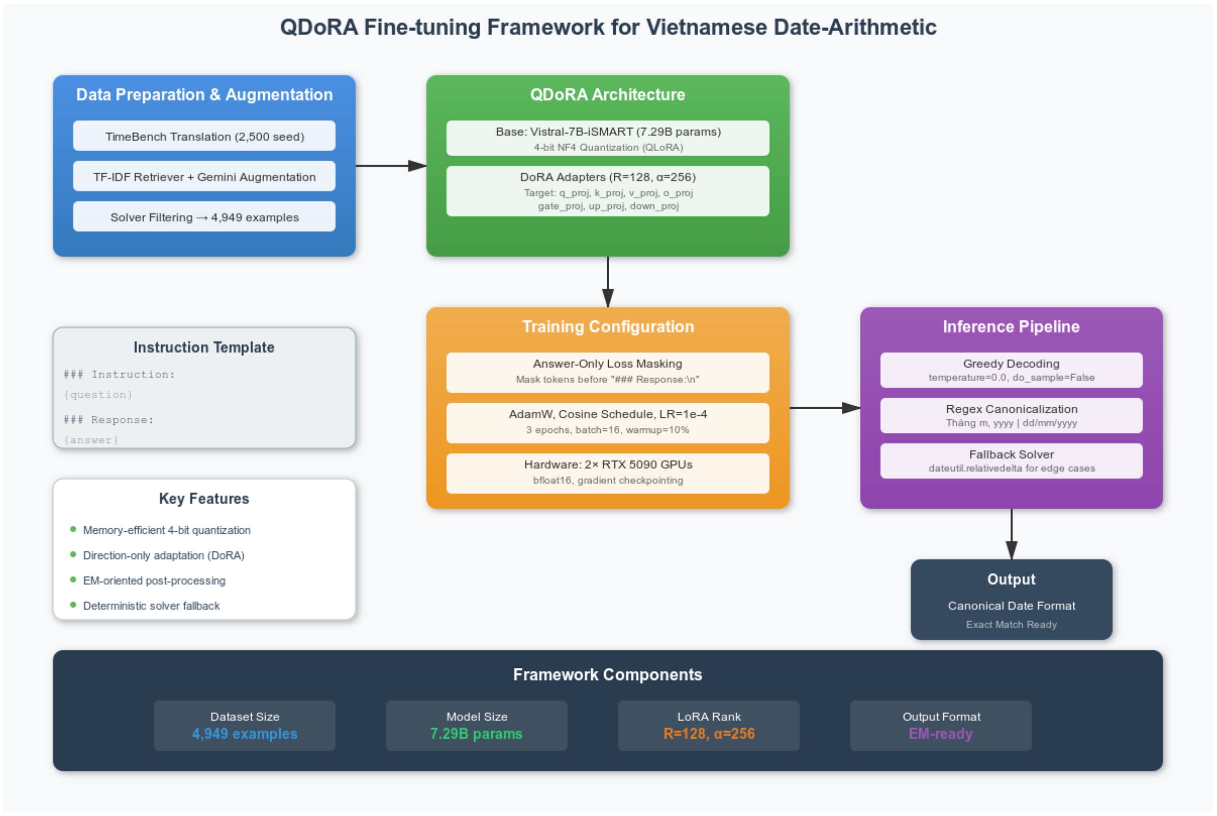
QDoRA Fine-tuning Framework for Vietnamese Date-Arithmetic
Data Preparation & Augmentation
TimeBench Translation (2,500 seed)
TF-IDF Retriever + Gemini Augmentation
Solver Filtering → 4,949 examples
QDoRA Architecture
Base: Vistral-7B-iSMART (7.29B params)
4-bit NF4 Quantization (QLoRA)
DoRA Adapters (R=128, α=256)
Target: q_proj, k_proj, v_proj, o_proj
gate_proj, up_proj, down_proj
Instruction Template
### Instruction:
{question}
### Response:
{answer}
Training Configuration
Answer-Only Loss Masking
Mask tokens before "### Response:\n"
AdamW, Cosine Schedule, LR=1e-4
3 epochs, batch=16, warmup=10%
Hardware: 2× RTX 5090 GPUs
bfloat16, gradient checkpointing
Inference Pipeline
Greedy Decoding
temperature=0.0, do_sample=False
Regex Canonicalization
Tháng m, yyyy | dd/mm/yyyy
Fallback Solver
dateutil.relativedelta for edge cases
Key Features
Memory-efficient 4-bit quantization
Direction-only adaptation (DoRA)
EM-oriented post-processing
Deterministic solver fallback
Output
Canonical Date Format
Exact Match Ready
Framework Components
Dataset Size
4,949 examples
Model Size
7.29B params
LoRA Rank
R=128, α=256
Output Format
EM-ready

Figure 1: An overview of our end-to-end framework, from data preparation and QDoRA architecture to the training and inference pipelines. Each component is optimized to produce a canonical, EM-ready output for the Vietnamese date-arithmetic task.

reduces sensitivity to quantization noise (which predominately perturbs magnitude), stabilizing optimization at higher ranks.

Algorithm 1 Training Pipeline

```
**Tokenization.** Truncate or pad all samples to 512 tokens.
**Masking.** In each sample, locate the ### Response: sentinel; set labels to $-100$ for all tokens appearing before it.
**Forward Pass.** Run the 4-bit quantized base model with DoRA adapters and compute the masked cross-entropy loss.
**Update.** Use the AdamW (torch-fused) optimizer with a cosine schedule and 10% warmup.

Repeat for 3 epochs, using gradient accumulation and gradient checkpointing.
```

Hyperparameters. A complete list of hyperparameters can be found in Table 1 in the Appendix.

Effective batch and steps. With two GPUs, the effective global batch size per optimizer step is:

$$\text{batch} = (\text{per-device} = 2) \times (\text{grad-acc} = 8) \times (\#\text{GPUs} = 2) = 32.$$

Given $N = 4,949$ examples, the number of steps per epoch is approximately 155, for a total of approximately 465 steps over 3 epochs.

3.2.3 Inference & EM-Oriented Post-processing

The inference pipeline consists of four stages:

- **Prompting (aligned to training).** We use a prompt identical to the training format:

  ### Instruction: {question}
  ### Response:

- **Decoding.** We use greedy decoding (do_sample=False, temperature=0.0) to avoid stochastic formatting drift and ensure deterministic outputs.
- **Canonicalization.** The raw model output is normalized to exactly one of two formats:

  - Tháng m, yyyy (for month-granularity anchors)
  - dd/mm/yyyy (for day-granularity anchors; zero-padded dd/mm)

  Any tokens that appear after a stray ### marker are stripped.

- **Fallback solver (deterministic).** If the generated string fails to match either regex, we apply a rule-based solver (using dateutil.relativedelta) directly to the input question. This solver computes the date based on calendar rules (month lengths, leap years, carry/borrow) without using any labels. The result is then reformatted canonically. This fairly rescues logically correct predictions that were merely formatted incorrectly.

Algorithm 2 Inference Pipeline

```
Build the instruction-aligned prompt for the input question.
Perform greedy decoding to generate a short completion ($\leq 24$ new tokens).
Apply regex normalization to convert the output to a canonical form.
if normalization fails then
    Call the deterministic fallback solver on the question.
    Reformat the solver's output to the canonical form.
end if
Emit the final canonical date string.
```

3.2.4 Parameter Settings (Concise)

A comprehensive summary of our hyperparameter and model configuration settings is provided in Table 1 in the Appendix.

3.2.5 Machine Specs and Processing Time

- **GPUs:** $2\times$ NVIDIA RTX 5090 (bf16 supported).
- **Parallelism:** DataParallel/Distributed, gradient checkpointing; device_map="auto".
- **CPU/RAM/OS:** Standard Linux server; exact CPU & RAM were not recorded.
- **Runtime:** 3h 25m.

4 Experimental Results

4.1 Evaluation Setup

We follow the official VLSP 2025 Temporal QA protocol for the Date Arithmetic sub-task. Two held-out sets are provided: a public test set for development and a private test set for final leaderboard evaluation. The primary metric is Exact Match (EM), meaning a prediction is correct only if the canonicalized output string exactly matches the gold date format (either dd/mm/yyyy or the Vietnamese "Tháng m, yyyy" format). Unless otherwise noted, all analyses below use the public test set and are run with greedy decoding (temperature 0) under identical hardware and decoding settings.

4.2 Dataset Sizes

The Public test set size is $N = 1,200$; Private test size is $N = 3,000$. We additionally construct several robustness subsets of size 300 each (see the Robustness section) and cross-lingual evaluation sets (see Cross-lingual positioning), described later.

4.3 Main Results on the Official Benchmark

Our submitted system is **Vistral-7B-iSMART**, fine-tuned with QDoRA and decoded with an EM-oriented pipeline (output canonicalization plus a deterministic solver fallback). On the official benchmark, the model attains **98.0% EM on the public test and 99.0% EM on the private test**, ranking first on the final leaderboard. The official results are summarized in Table 2 in the Appendix. These results indicate that parameter-efficient fine-tuning combined with form-constrained decoding closes most of the gap on symbolic temporal reasoning in Vietnamese date arithmetic.

4.4 Component Analyses (Ablations)

We quantify the contribution of each component by varying one factor at a time relative to the final system ("Final"). We evaluate variants that remove individual components (solver fallback, answer-only loss masking, output canonicalization), replace QDoRA with standard LoRA under identical 4-bit settings, and vary the adapter rank $R \in \{16, 64, 96, 128\}$ (with the LoRA scaling factor α adjusted proportionally) to study the quality–efficiency trade-off. All results are reported as EM (%) on the public test unless stated otherwise; we also report Relaxed EM (logical equivalence after normalization), decoding throughput (tokens/s), and peak GPU memory usage. The full results are presented in Appendix Table 3.

Insights.

(i) **Canonicalization and solver fallback are the largest EM drivers** (adding +3.5 and +2.3 points, respectively, when present). The gap between EM and Relaxed EM without canonicalization (see Appendix Table 3) shows that many "errors" are formatting-only; canonicalizing the output converts those into exact matches.

(ii) **Answer-only masking contributes about +1.2 EM**, presumably by reducing objective mismatch and focusing learning on the answer span.

(iii) Under 4-bit training, QDoRA outperforms standard LoRA at the same rank ($R = 96$) by **+0.8 EM**, reflecting improved expressiveness and stability from the weight decomposition.

(iv) Adapter rank has diminishing returns: There is a practical sweet spot around $R \approx 96$. For example, increasing to $R = 128$ yields only a marginal +0.1 EM gain at the cost of higher memory usage and slightly lower throughput.

Statistical testing: For the three largest drops (removing canonicalization, fallback, or masking), McNemar's test comparing the variant to the Final model's predictions yields $p < 0.01$ in each case, indicating those differences are statistically significant.

4.5 Robustness to Linguistic Variation and Boundary Conditions

We construct five controlled robustness test sets ($N = 300$ each) that introduce realistic input perturbations: R1 – spelling/punctuation noise; R2 – abbreviations or synonyms for temporal operators (e.g. "trc/sau" for before/after); R3 – clause reordering; R4 – textual numerals (e.g. "mười hai", "hai mươi ba"); and R5 – boundary stress (leap years, end-of-month dates, year crossings). We evaluate the Final model on these sets and also compare to a standard LoRA baseline ($R = 96$). The results for both models are reported in Table 4 in the Appendix.

Insights.

(i) **Boundary cases (R5) are the most challenging**, with EM dropping to 94.2% (-3.8 points relative to the clean test set). This is consistent with known difficulties in end-of-month carry/borrow and leap-year edge cases.

(ii) Textual numerals (R4) remain a source of brittleness. However, the high Relaxed EM (98.6%) suggests that many predictions in this category are logically correct but misformatted due to numeral parsing artifacts. In other words, the model often gets the right date but expresses it incorrectly when numerals are written out as words.

(iii) Across all perturbations, the Final QDoRA model maintains roughly a **0.7–1.5 EM point advantage over the LoRA baseline**, indicating that QDoRA's accuracy gains persist under distribution shift.

4.6 Cross-Lingual Positioning on English Symbolic Subsets

To probe transferability beyond Vietnamese, we evaluate our approach on an English symbolic dataset derived from TIMEBench ($N = 800$). We consider two deployment variants: (i) an EN$\rightarrow$VI translation front-end, where each English query is machine-translated to Vietnamese and answered by our Vietnamese model (with the result then converted to the English date format via canonicalization); and (ii) a native English parser that directly extracts the temporal anchor and the signed duration from the English query, feeding these to the same solver and then applying English output canonicalization. We also include small illustrative samples from two English temporal reasoning benchmarks focused on commonsense and events (TORQUE and MC-TACO, $N = 300$ each) to delineate the scope of our approach on non-symbolic temporal questions. Our findings from this evaluation are presented in Table 5 in the Appendix.

Insights.

(i) **High EM on the English symbolic subset** ($\geq 96.9\%$) indicates that our EM-oriented decoding pipeline and solver-verified approach are language-agnostic for purely symbolic temporal arithmetic. The system retains nearly the same accuracy with English queries (either via translation or direct parsing) as it does on Vietnamese.

(ii) The marked performance drop on TORQUE and MC-TACO confirms that **commonsense**

or event-centric temporal questions require reasoning beyond pure symbolic computation. Our method excels at deterministic date arithmetic, but integrating world knowledge and contextual event reasoning (as needed for TORQUE/MC-TACO) remains an open challenge – these results clearly demarcate the scope of our approach.

4.7 Error Analysis

We manually inspect $N = 100$ residual errors from the public test and robustness sets, and we categorize each error into one of five types (E1–E5), as summarized in Table 6 in the Appendix.

Representative observations.

(i) **Boundary errors (E2)** often arise from month-end carry/borrow issues and the special case of February in leap vs. non-leap years. Incorporating explicit "end-of-month" logic or training examples could reduce this category significantly.

(ii) Formatting errors (E4) are largely eliminated by our post-processing canonicalizer, but a few failures remain – for example, when the model correctly computes the date but then emits extra tokens or an incorrect format beyond the date string. Implementing stricter format constraints in decoding or post-filtering any non-date tokens can help address these cases.

(iii) **Vietnamese parsing ambiguities (E5)** mainly involve overlapping numeral interpretations. For instance, the phrase "mười hai" (12) can overlap with "hai" (2) if the system naively scans for digit words, leading to misinterpretation. A more robust, non-overlapping numeral parsing strategy (e.g. always matching the longest valid numeral phrase) would mitigate these errors.

4.8 Efficiency and Memory Footprint

We measure inference throughput and memory usage for different model variants under identical hardware and decoding parameters (greedy decoding with max_new_tokens=24, batch size 1 for latency). All results are obtained on a single NVIDIA RTX 5090, and training was performed on 2× NVIDIA RTX 5090s. Detailed efficiency metrics are provided in Table 7 in the Appendix.

Insights.

(i) **EM-oriented decoding adds negligible overhead**. Our pipeline's extra steps (canonicalization and occasional solver calls) have minimal impact on latency. The fallback solver is invoked for $\leq 3\%$ of queries thanks to high direct canonicalization success, and even when used, the solver's runtime is small compared to generation latency.

(ii) Increasing adapter rank from 96 to 128 has diminishing returns: it yields only +0.1 EM (as noted earlier) while reducing throughput (204 vs 210 tokens/s) and increasing memory usage. This suggests that $R \approx 96$ **is a good Pareto optimal point** for this task, beyond which gains are marginal.

(iii) **LoRA and QDoRA have similar runtime profiles**. The quantized QLoRA approach does not incur additional inference cost relative to standard LoRA at the same rank. The primary benefit of QDoRA is improved accuracy under 4-bit quantization, not speed – as shown, both LoRA and QDoRA achieve ~ 210 tokens/s and have comparable latency.

5 Discussion

We synthesize the empirical findings to explain why the proposed system attains near-ceiling EM on Vietnamese date arithmetic, which components matter most, and where the remaining failure modes lie. We also outline practical considerations and limitations, together with concrete next steps.

5.1 Why QDoRA for date arithmetic

Direction-only adaptation under 4-bit. Direction-only adaptation under 4-bit means DoRA decomposes each pretrained weight into magnitude and direction and updates only the direction. Under 4-bit QLoRA, magnitude is the dimension most sensitive to quantization error; freezing it stabilizes optimization—especially at higher adapter ranks—while preserving pretrained priors useful for rule-like reasoning. Empirically (see Appendix Table 3), QDoRA@R=96 outperforms LoRA@R=96 by +0.8 EM with identical hyperparameters, and remains robust across ranks.

Closer to full-FT behavior on structured logic. Date arithmetic is structured and compositional (carry/borrow across months, leap-year rules). Directional updates permit coherent rotations in attention/MLP subspaces, approximating full fine-tuning with $\approx$ 1–2% trainable parameters and no inference-time latency. The small—but consistent—edge at higher ranks ($R = 128$ vs. $R = 96$: +0.1 EM) mirrors this behavior, albeit with diminishing returns and higher VRAM (see Appendix Tables 3 and 7).

5.2 Answer-only loss masking

EM is assessed solely on the completion span. Masking out all tokens before the ### Response: sentinel removes gradients from instruction boilerplate, focusing capacity on the output subspace. Ablation confirms a -1.2 EM drop when masking is disabled (see Appendix Table 3), indicating reduced objective mismatch and fewer stylistic artifacts in the decoded string.

5.3 EM-oriented decoding and canonicalization

Greedy decoding avoids sampling noise that harms exact string match. Regex canonicalization collapses benign surface variation (whitespace, hyphen vs. slash) into the two allowed formats. Removing canonicalization yields the single largest drop (-3.5 EM, see Appendix Table 3) while Relaxed EM remains 99.1, showing most "errors" are formatting-only. The deterministic solver fallback corrects rare residual formatting failures without labels, contributing +2.3 EM (Final vs. – Fallback). Jointly, these steps convert logically correct reasoning into exact-match strings—precisely what the metric rewards.

5.4 Data augmentation with solver filtering

RAG-prompted variants are filtered by a deterministic solver; only logically valid items are retained (1,949 kept; total 4,949). This guarantees target consistency and densifies tail cases (end-of-month, leap year, multi-unit offsets) that disproportionately affect EM. Robustness results (see Appendix Table 4) suggest that the training distribution sufficiently exposes boundary phenomena: the Final model degrades modestly under noise and remains resilient relative to the LoRA baseline.

5.5 Why these components work together

- **Capacity & stability:** QDoRA supplies stable, high-rank capacity to internalize symbolic rules under 4-bit compression.
- **Objective alignment:** Answer-only masking ensures that capacity is spent on the exact region EM evaluates.
- **Form-faithful decoding:** Greedy + canonicalization + fallback systematically convert "right-in-logic" outputs into "right-as-strings."
- **Distribution shaping:** Solver-filtered augmentation targets failure modes that matter (EOM, leap-year, multi-unit deltas), improving generalization where EM is most brittle.

The net effect is visible across analyses: removing canonicalization or fallback collapses EM despite high Relaxed EM (see Appendix Table 3), while QDoRA's gains persist under perturbations (see Appendix Table 4).

5.6 Practical considerations and complexity

Memory. 4-bit base storage with bf16 adapters (and optimizer states for adapters only) keeps VRAM modest. Peak $\text{VRAM}_{\text{train}}$ for the Final model is 29.1 GB/GPU, and $\text{VRAM}_{\text{infer}}$ is 8.6 GB (see Appendix Table 7). Gradient checkpointing limits activation memory. **Time.** With $N = 4,949$, global batch size 32, and 3 epochs, the training loop runs ≈ 465 updates. Wall-clock for the Final model is 3 h 25 m on 2$\times$ GPUs. **Latency.** Inference sustains $\approx$ 210 tokens/s with median 120 ms / P95 180 ms per query. The fallback solver is invoked on $\leq 3\%$ of queries and does not dominate latency. **Sweet spot.** $R = 128$ yields only +0.1 EM over $R = 96$ but reduces throughput and increases memory (see Appendix Table 7), making $R \approx 96$ a pragmatic Pareto point for this task.

6 Conclusion

We presented a champion system for the VLSP 2025 Temporal QA Date Arithmetic sub-task that combines parameter-efficient fine-tuning with form-constrained decoding. Fine-tuning **Vistral-7B-iSMART** via QDoRA and decoding with an EM-oriented pipeline (greedy search, regex canonicalization, deterministic solver fallback) yields **98.0%** EM on the public test and **99.0%** EM on the private test, topping the leaderboard. Beyond headline numbers, our results demonstrate that symbolic temporal reasoning in Vietnamese can be brought close to ceiling with carefully aligned modeling, training, and decoding choices.

Our empirical analysis clarifies **which components matter** and **why**. Ablations show that canonicalization and solver fallback are the largest drivers of EM (-3.5 and -2.3 EM when removed), converting logically correct predictions into exact matches. **Answer-only masking** reduces objective mismatch (-1.2 EM when removed), and **QDoRA** outperforms standard LoRA under the same 4-bit setting (+0.8 EM at R=96), indicating improved expressiveness and stability for rule-like computation. Rank sensitivity reveals a pragmatic **sweet spot near** $R \approx 96$: $R = 128$ adds only +0.1 EM while increasing VRAM and lowering throughput. The system is **fast and compact** ($\approx$ 210 tokens/s, 8.6 GB VRAM inference; 29.1 GB VRAM/GPU training), making it practical for deployment.

We also evaluated **robustness** and **transferability**. Under realistic perturbations—spelling/punctuation noise, abbreviations/synonyms, clause reordering, textual numerals, and boundary stress—the model remains strong (e.g., EM 97.3 under R1, 95.9 under R4), with **Relaxed EM** indicating that many failures are formatting-only. On an **English symbolic** subset, both an EN$\rightarrow$VI front-end and a native English parser retain high EM (97.5/96.9), suggesting language-agnostic applicability for deterministic date arithmetic. In contrast, performance drops on **commonsense/event** benchmarks (e.g., TORQUE 47.3 EM; MC-TACO 56.2 Acc), delineating the boundary between symbolic arithmetic and knowledge-heavy temporal reasoning.

Our **limitations** follow directly from this analysis. First, the approach is **metric-centric**: EM rewards form correctness, and gains partly arise from post-hoc canonicalization. Second, residual errors concentrate in **boundary conditions** (end-of-month/leap-year; 34% of analyzed errors) and **textual numerals** (18%), reflecting known brittleness in month-end carry/borrow and overlapping numeral spans. Third, generalization to semantics-heavy temporal QA remains incomplete, as evidenced by the gap on TORQUE/MC-TACO.

These observations open several **avenues for future work**. On decoding and parsing, we will (i) adopt **constrained decoding** to eliminate trailing tokens and enforce date formats at generation time; (ii) implement **longest-match, non-overlapping** numeral parsing with unit-aware grammars; and (iii) encode **end-of-month semantics** directly in training (via augmentation or rule-regularized objectives). On reasoning, we will (iv) extend beyond symbolic arithmetic by **integrating retrieval over temporal KBs/corpora**, and (v) explore **neuro-symbolic** pipelines that apply solver-style checks to event timelines. On evaluation and openness, we plan to (vi) release splits, augmentation templates, and code to strengthen **reproducibility**, and (vii) broaden multilingual experiments to further test cross-lingual stability.

In sum, the proposed **QDoRA-fine-tuned, EM-oriented** system offers a principled, compute-efficient recipe for **high-precision symbolic temporal QA** in Vietnamese. By aligning objective, capacity, and decoding with the target metric, we achieve near-ceiling EM while maintaining speed and memory efficiency. The analysis pinpoints precisely what remains hard—boundary semantics and temporal commonsense—charting a clear path toward next-generation temporal reasoning systems that combine symbolic reliability with semantic breadth.

References

Association for Vietnamese Language and Speech Processing. 2025. VLSP 2025 challenge on temporal QA. https://vlsp.org.vn/vlsp2025/eval/tempqa.

Zhaocheng Chu, Jiali Chen, Qidong Chen, Wentao Yu, Hao Wang, Ming Liu, and Bing Qin. 2024. TIMEBENCH: A comprehensive evaluation of temporal reasoning abilities in large language models. *Preprint*, arXiv:2311.17667.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. *Preprint*, arXiv:2305.14314.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Patrice Lambert and Kiem-Hieu Nguyen. 2012. Recognition of vietnamese time expressions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(4).

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-decomposed low-rank adaptation. *Preprint*, arXiv:2402.09353.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. TORQUE: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Jannik Strötgen and Kiem-Hieu Nguyen. 2014. Extending HeidelTime for Vietnamese. In *Proceedings of the 1st International Workshop on Vietnamese Language and Speech Processing (VLSP).*

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).*

A Tables

Component	**Setting**
Base model	Vistral-7B-iSMART (7B)
Adapters	DoRA over QLoRA base
Target modules	q/k/v/o_proj; gate/up/down_proj
Rank / α / dropout	$R = 128$, $\alpha = 256$, $p = 0.05$
Quantization	4-bit NF4, double-quant
Sequence length	512
Optimizer	AdamW (torch-fused)
LR / schedule	10^{-4}, cosine, warmup 10%
Epochs	3
Per-device batch	2
Grad accumulation	8
Global batch (2 GPUs)	32
Compute dtype	bf16 (fallback fp16)
Seed	42

Table 1: Hyperparameter and model configuration settings.

Table 2: Official VLSP 2025 Date Arithmetic results (EM %)

Test Set	EM (%)	N
Public test	98.0 (95% CI 97.2–98.8)	1,200
Private test (official)	99.0 (95% CI 98.6–99.4)	3,000

Table 3: Ablation summary on public test (N=1,200)

Variant	EM	Relaxed EM	ΔEM vs Final	tokens/s	VRAM$_{\text{train}}$	VRAM$_{\text{infer}}$
Final (QDoRA, R=96)	98.0	99.2	–	210	29.1 GB/GPU	8.6 GB
– Fallback solver	95.7	99.0	−2.3	215	29.1 GB	8.4 GB
– Answer-only masking	96.8	97.4	−1.2	210	29.1 GB	8.6 GB
– Canonicalization	94.5	99.1	−3.5	220	29.1 GB	8.3 GB
LoRA (no DoRA), R=96	97.2	98.4	−0.8	211	28.7 GB	8.5 GB
QDoRA, R=16	96.1	97.6	−1.9	213	27.3 GB	8.2 GB
QDoRA, R=64	97.6	98.8	−0.4	211	28.5 GB	8.5 GB
QDoRA, R=128	98.1	99.2	+0.1	204	30.8 GB	8.9 GB

Table 4: Robustness results (N=300 each)

Noise/Perturbation Type	Final EM	Final Relaxed EM	LoRA EM
R1: Spelling/punctuation noise	97.3	99.0	96.4
R2: Abbrev./synonyms (trc/sau)	96.8	98.7	95.9
R3: Clause reordering	97.1	99.1	96.2
R4: Textual numerals	95.9	98.6	94.8
R5: Boundary stress (EOM/leap)	94.2	96.9	93.4

Table 5: Cross-lingual symbolic and illustrative commonsense/event evaluation

Evaluation Set	Metric	Score
TIMEBench-EN (symbolic) – EN→VI model pipeline	EM	97.5
TIMEBench-EN (symbolic) – Native EN parser	EM	96.9
TORQUE sample (event ordering questions)	EM	47.3
MC-TACO sample (commonsense temporal questions)	Acc	56.2

Table 6: Error taxonomy (share of errors)

Error Type		Share of Errors (%)
E1.	Offset miscalculation	18
E2.	Boundary handling (end-of-month/leap-year)	34
E3.	Anchor mismatch (day vs month vs year)	9
E4.	Formatting failures	21
E5.	Vietnamese parsing issues (textual numerals, etc.)	18

Table 7: Efficiency metrics

Model/Variant	Tokens/s	Median latency (ms)	P95 latency (ms)	VRAM$_{\text{infer}}$	VRAM$_{\text{train}}$ (per GPU)
Final (QDoRA, R=96)	210	120	180	8.6 GB	29.1 GB
– Fallback solver	215	117	175	8.4 GB	29.1 GB
LoRA (R=96)	211	121	182	8.5 GB	28.7 GB
QDoRA (R=128)	204	126	189	8.9 GB	30.8 GB