FAITHFUL EVIDENCE BY CONSTRUCTION FOR TEMPORAL KNOWLEDGE GRAPH REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal knowledge graph question answering (TKGQA) requires natural language understanding together with explicit temporal reasoning over facts annotated with timestamps or intervals. State-of-the-art methods follow a *retrieve-then-reason* paradigm that gathers evidence by dense retrieval and delegates reasoning to repeatedly invoked large language models (LLMs), which compounds two failure modes: structurally and temporally unfaithful evidence sets, and unbounded error amplification across multiple LLM calls. We propose **FaiTH**, a training-free framework organized as a four-stage compile, ground, execute, and reason pipeline. One LLM call compiles the question into a dataset-agnostic specification, a deterministic cascaded grounder resolves mentions to canonical identifiers, a symbolic engine performs role-aware retrieval, granularity-aware temporal filtering, inter-event alignment, and multi-event composition to produce a provably faithful evidence set, and an adaptive reasoner resolves residual ambiguity. Across three heterogeneous TKGQA benchmarks, FaiTH achieves state-of-the-art Hits@1 scores, reaching 0.852 on MULTITQ against 0.780 for the strongest baseline, using at most two LLM calls to answer each question. FaiTH's code and data are anonymously available at `https://anonymous.4open.science/r/FaiTH-978E/`.

1 INTRODUCTION

Temporal knowledge graphs (TKGs) extend static knowledge graphs (Bollacker et al., 2008; Vrandečić & Krötzsch, 2014) by giving every fact a timestamp or a validity interval (Lacroix et al., 2020), so temporal knowledge graph question answering (TKGQA) (Jia et al., 2018; Saxena et al., 2021; Su et al., 2024) demands explicit reasoning over constraints such as "before", "first", and "during", often composed across facts on different time scales (Chen et al., 2023; Allen, 1983). Early methods score answers with embeddings (Saxena et al., 2021; 2020; Mavromatis et al., 2022; Chen et al., 2023) or graph neural networks (Jia et al., 2021; Sharma et al., 2023; Liu et al., 2023b;a; Zha et al., 2024); recent work recasts TKGQA as generation under a *retrieve-then-reason* paradigm (Lewis et al., 2020; Chen et al., 2024b; Qian et al., 2024; 2025; Gong et al., 2025; 2026) and holds the state of the art (SOTA). This paradigm carries two failure modes that limit reliability and efficiency.

The first is **structural and temporal unfaithfulness of the retrieved evidence**. Dense retrievers rank facts by embedding similarity, which ignores the semantic role an entity plays: a query about e *investigating* a target as a *subject* may equally return facts where e is the *object* (Luo et al., 2025), and temporal constraints are encoded softly rather than enforced. The LLM then cannot separate faithful evidence from noise and tends to conflate roles or misread temporal cues (Chen et al., 2024a). PoK (Qian et al., 2025) reports an answer coverage of 71.78% at top-50 on MULTITQ (Chen et al., 2023), so on more than a quarter of its questions no retrieved fact carries the gold answer.

The second is **unfaithfulness amplification through repeated LLM invocations**. Iterative refinement inside the LLM (Wang et al., 2023b) is the natural remedy for imperfect retrieval, yet every call adds hallucination risk (Huang et al., 2025) that propagates into later rounds (Wang et al., 2023a). RTQA (Gong et al., 2025) averages 3.96 calls per question and ReAct-style agents (Yao et al., 2022; QianyiHu et al., 2025; Tan et al., 2025) can use more, and no prior TKGQA method certifies that every fact in its evidence set is structurally and temporally faithful to the question.

We propose **FaiTH** (**Fai**thful **T**emporal knowledge graph reasoning for **H**eterogeneous time formats), a training-free framework for both failure modes, where the **H**eterogeneity lies in point-stamped

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

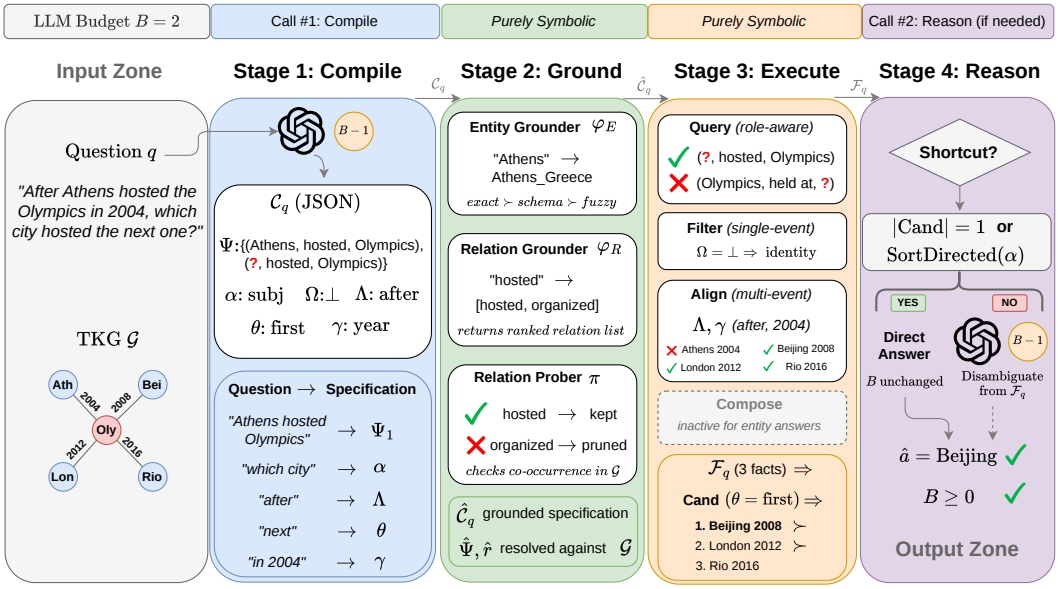

Figure 1: Overview of FaiTH. Stage 1 compiles q into a constraint specification $\mathcal{C}_q$ with one LLM call; Stage 2 grounds every mention and Stage 3 runs four symbolic operators, both without the LLM, yielding a faithful evidence set $\mathcal{F}_q$ and a candidate list; Stage 4 shortcuts when the candidate is unambiguous and otherwise consults the LLM once, so the budget never exceeds two calls.

events (Chen et al., 2023), interval-valued facts, and sentinel endpoints (Boschee et al., 2015; Sun et al., 2025; Saxena et al., 2021). FaiTH compiles a question once into a typed specification, grounds and executes it symbolically, and calls the LLM again only when ambiguity remains. We formalize structural and temporal faithfulness, prove that the symbolic operators return an evidence set satisfying both, and evaluate 11 LLM backbone configurations spanning 9 base models on three benchmarks.

2 Related Work

Embedding and GNN methods. Early methods cast TKGQA as TKG completion and score answers by embedding similarity, as in CronKGQA (Saxena et al., 2021; Lacroix et al., 2020; Devlin et al., 2019), EmbedKGQA (Saxena et al., 2020; Trouillon et al., 2016), TempoQR (Mavromatis et al., 2022), and MultiQA (Chen et al., 2023; Vaswani et al., 2023), while graph neural networks model structure explicitly, as in EXAQT (Jia et al., 2021), TwiRGCN (Sharma et al., 2023), LGQA (Liu et al., 2023b), TMA (Liu et al., 2023a), and M3TQA (Zha et al., 2024). Both families stay tied to the answer vocabulary of the graph and degrade on multi-hop composition.

LLM-based methods. Prompt-engineered pipelines place a closed-source LLM inside a fixed workflow, as in ARI (Chen et al., 2024b), the ReAct-style TempAgent (QianyiHu et al., 2025; Yao et al., 2022), MemoTime (Tan et al., 2025), and RTQA (Gong et al., 2025). Retrieval-augmented methods pair retrievers with fine-tuned open-source models, as in TimeR^4 (Qian et al., 2024; Reimers & Gurevych, 2019; Touvron et al., 2023; OpenAI, 2023) and PoK (Qian et al., 2025; OpenAI, 2024a; Rusak et al., 2025), and reinforcement learning gives agents an explicit action space, as in Temp-R1 (Gong et al., 2026; Grattafiori et al., 2024; Guo et al., 2025) and Search-R1 (Jin et al., 2025). None of them verifies that the evidence shown to the LLM is faithful to the question.

Constrained and symbolic KGQA. Faithful LLM reasoning over KGs uses constrained decoding and graph-guided search (Luo et al., 2025; Chen et al., 2024a), and semantic-parsing KGQA compiles questions into logical forms over static KGs, inspiring Stage 1. Those forms lack the interval arithmetic, granularity truncation, sentinels, and Allen relations that FaiTH executes (Appendix A).

3 PRELIMINARIES AND PROBLEM SETTING

A temporal knowledge graph (TKG) is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{T}$ denote the sets of entities, relations, and timestamps, and $\mathcal{F}$ is the set of temporal facts. Each fact is either a *point-valued* quadruple (s, r, o, τ) or an *interval-valued* quintuple $(s, r, o, \tau_s, \tau_e)$ with validity interval $[\tau_s, \tau_e] \subseteq \mathcal{T}$, and Section 4.1 unifies the two. Given $\mathcal{G}$ and a natural language question q, temporal knowledge graph question answering (TKGQA) infers the answer a from $\mathcal{F}$; a may be an entity, a timestamp or interval, a duration, a ranking index, or the result of a set operation on intervals. Figure 1 shows the four-stage *compile-ground-execute-reason* pipeline that follows; Appendix B.1 motivates the design, Table 5 lists every symbol, and Appendix B.3 covers instantiation.

4 THE FAITH FRAMEWORK

4.1 STAGE 1: STRUCTURAL COMPILATION

Stage 1 converts q into a compact, dataset-agnostic *constraint specification* for all later stages.

Unified fact and time domain. We treat every fact as a quintuple $(s, r, o, \tau_s, \tau_e)$, reading a point-valued quadruple (s, r, o, τ) as the degenerate interval (s, r, o, τ, τ), and embed $\mathcal{T}$ into an extended domain that admits open-ended intervals,

$$\mathcal{T}^\star \;=\; \mathcal{T} \cup \{\tau_{-\infty}, \tau_{+\infty}\}, \tag{1}$$

with the total order $\preceq$ placing $\tau_{-\infty}$ below and $\tau_{+\infty}$ above all of $\mathcal{T}$. Sentinels such as *beginning of time* and *end of time* map to $\tau_{-\infty}$ and $\tau_{+\infty}$, while point-valued benchmarks keep $\mathcal{T}^\star = \mathcal{T}$.

Constraint specification. Given a question q, we compile it into a *FaiTH specification*

$$\mathcal{C}_q \;=\; \big(\,\Psi,\ \alpha,\ \Omega,\ \Lambda,\ \theta,\ \gamma\,\big), \tag{2}$$

whose six components encode *what* the question asks, *where* the unknown lies, and *under which temporal conditions* the answer must hold. The *event templates* $\Psi = \{\psi_i\}_{i=1}^{k}$ are $k \in \{1, 2, 3\}$ triples $\psi_i = (s_i, r_i, o_i)$ matching the fact schema of $\mathcal{G}$, each with at most one slot marked unknown as "?", and the *answer slot* α names what the question returns, drawn from

$$\mathcal{A} \;=\; \{\,\mathsf{subj},\ \mathsf{obj},\ \mathsf{tss},\ \mathsf{tse},\ \mathsf{tsr},\ \mathsf{dur},\ \mathsf{rdur},\ \mathsf{rank},\ \mathsf{setop}\,\}, \tag{3}$$

covering, in order, subject entity, object entity, start timestamp, end timestamp, full interval, single-event duration, aggregated duration, ranked position, and set-operation result; point-valued benchmarks exercise only $\{\mathsf{subj}, \mathsf{obj}, \mathsf{tss}, \mathsf{tse}\}$ and interval benchmarks all nine. The *single-event temporal constraint* $\Omega = [\tau_s^q, \tau_e^q] \subseteq \mathcal{T}^\star$ confines the target fact to a window, with a null Ω meaning no absolute time constraint, and the *inter-event relation* $\Lambda = \{\lambda_{ij}\}$, active only when $k \geq 2$, assigns $\lambda_{ij} \in \mathcal{L}$ between pairs of templates, where

$$\mathcal{L} \;=\; \{\,\mathsf{before},\ \mathsf{after},\ \mathsf{during},\ \mathsf{sametime},\ \mathsf{finishsame},\ \mathsf{startsame}\,\} \tag{4}$$

projects the thirteen Allen relations (Allen, 1983) onto six predicates over interval pairs (Appendix B.4). The *temporal operator* θ selects the timeline arithmetic and carries any ordering directive, and $\gamma \in \{\mathsf{day}, \mathsf{month}, \mathsf{year}\}$ sets the granularity of all comparisons.

Compilation procedure. A single LLM call with a few-shot prompt implements $q \mapsto \mathcal{C}_q$ and returns JSON, which two deterministic steps repair: a *rule-based repair* recovers temporal connectives from surface cues such as "before" or "first" when the LLM leaves Λ or an ordering directive empty, and a *passive-voice role swap* corrects the unknown's role for predicates the compiler systematically mis-locates. Repair overrides LLM output, so a deterministic signal in q always wins, and this call is charged once to the budget B (prompt in Appendix D).

4.2 STAGE 2: SYMBOL GROUNDING

The templates of $\mathcal{C}_q$ name entities and relations by free surface forms that need not coincide with canonical identifiers in $\mathcal{G}$. Stage 2 resolves every mention into a *grounded specification* that Stage 3 can execute symbolically, and consumes no LLM budget.

Grounding operator. Φ updates Ψ to $\hat{\Psi}$ and leaves the other five fields untouched. Let $\mathcal{M}$ be the set of surface strings. A deterministic entity grounder $\phi_E : \mathcal{M} \to \mathcal{E} \cup \{\perp\}$, with $\perp$ marking failure, and a relation grounder $\phi_R : \mathcal{M} \to 2^{\mathcal{R}}$ lift to

$$\Phi(\mathcal{C}_q, \mathcal{G}) \;=\; \hat{\mathcal{C}}_q \;=\; \big(\,\hat{\Psi},\; \alpha,\; \Omega,\; \Lambda,\; \theta,\; \gamma\,\big), \tag{5}$$

where $\hat{\Psi}$ replaces every non-placeholder slot of Ψ by its canonical entity identifier under ϕ_E and every relation by the ordered list $\phi_R(r_i) = [\hat{r}_{i,1}, \ldots, \hat{r}_{i,m}]$.

Confidence-ranked cascade. Both ϕ_E (with $c \in \mathcal{E}$) and ϕ_R (with $c \in \mathcal{R}$) run matching rules in sequence and return the candidate that fires the highest-confidence one,

$$\phi(m) \;=\; \operatorname{argmax}_c \;\; \max_{\ell=1}^{L} \; w_\ell \cdot \mathbf{1}[\, c \in \mu_\ell(m, \mathcal{G})\,], \tag{6}$$

where $\mu_1, \ldots, \mu_L$ have strictly decreasing precision and $w_1 > \cdots > w_L > 0$ are decreasing weights. The cascade starts from exact and case-insensitive string matching, moves through schema-aware transformations and morphological or synonym relaxations, and falls back to token-overlap and sequence-similarity scores. Only $\{\mu_\ell\}$ follows the naming convention of each TKG (Appendix B.3).

Relation probing. Pre-built indices discard candidate relations that never co-occur with the anchor entity in $\mathcal{G}$. Because relation grounding returns a list rather than one identifier, Φ applies a constant-time co-occurrence probe over the inverted indices,

$$\chi(\hat{e}, \hat{r}) \;=\; \mathbf{1}\big[\,\exists\,(s, r, o, \tau_s, \tau_e) \in \mathcal{F} \;\text{ s.t. }\; r = \hat{r} \,\wedge\, \hat{e} \in \{s, o\}\,\big], \tag{7}$$

evaluated through inverted indices, so $\hat{r}_{i,j}$ survives only if it passes χ against the anchor of its template. When every forward candidate fails, Φ retries inverse relations; when both fail the template stays ungrounded and Stage 3 surfaces an empty evidence set for the recovery of Appendix B.5.

4.3 Stage 3: Symbolic Constraint Execution

Stage 3 turns $\hat{\mathcal{C}}_q$ into a structurally and temporally consistent evidence set $\mathcal{F}_q \subseteq \mathcal{F}$ and a candidate list, entirely symbolically and with no LLM budget.

Definition 1 (Structural consistency)**.** *A fact $f = (s, r, o, \tau_s, \tau_e) \in \mathcal{F}$ is* structurally consistent *with a grounded template $\hat{\psi}_i = (\hat{s}_i, \hat{r}_i, \hat{o}_i) \in \hat{\Psi}$ if $r = \hat{r}_i$ and every non-placeholder slot of $\hat{\psi}_i$ coincides with the corresponding element of f; that is, if $s = \hat{s}_i$ whenever $\hat{s}_i \neq ?$, and $o = \hat{o}_i$ whenever $\hat{o}_i \neq ?$.*

Definition 2 (Temporal consistency)**.** *A fact $f = (s, r, o, \tau_s, \tau_e)$ is* temporally consistent *with $\hat{\mathcal{C}}_q$ at granularity γ if it satisfies the window $\lfloor \tau_s^q \rfloor_\gamma \preceq \lfloor \tau_s \rfloor_\gamma$ and $\lfloor \tau_e \rfloor_\gamma \preceq \lfloor \tau_e^q \rfloor_\gamma$ whenever $\Omega \neq \perp$, and, when $k \geq 2$, if every $\lambda_{ij} \in \Lambda$ holds between f and the envelope of the facts retrieved for the paired template, all comparisons under $\preceq$ on $\mathcal{T}^\star$.*

Definition 3 (Faithful evidence set)**.** *An evidence set $\mathcal{F}_q \subseteq \mathcal{F}$ is* faithful *to $\hat{\mathcal{C}}_q$ if every $f \in \mathcal{F}_q$ is structurally consistent with some $\hat{\psi}_i \in \hat{\Psi}$ and temporally consistent with $\hat{\mathcal{C}}_q$, and every fact of $\mathcal{F}$ satisfying both lies in $\mathcal{F}_q$ unless tied at a boundary whose tie group has filled the adaptive bound.*

The next three operators satisfy Definition 3 under the grounding assumption stated first.

Assumption 1 (Correct grounding)**.** *The grounding operator Φ resolves every non-placeholder mention in $\mathcal{C}_q$ to the canonical identifier the question intends: $\hat{e}_i \in \mathcal{E}$ is the intended anchor of template ψ_i, and the retained candidates $\hat{r}_{i,j} \in \mathcal{R}$ contain the intended relation.*

Assumption 1 separates the guarantees of the symbolic engine from grounding quality, which Appendix C.9 measures empirically; the two statements below are conditional on it and proved in Appendix B.6, and Appendix C.4 traces each definition through worked cases.

Role-aware retrieval. Let $\hat{\psi}_i = (\hat{s}_i, \hat{r}_i, \hat{o}_i)$ be a grounded template in which the unknown occupies role $\rho \in \{\mathsf{subj}, \mathsf{obj}\}$. Writing $\bar{\rho}$ for the complementary role and $\pi_{\bar{\rho}}(f)$ for the projection of fact f onto $\bar{\rho}$, the retrieval operator collects

$$\mathrm{Query}(\hat{\psi}_i, \rho) \;=\; \big\{\, f = (s, r, o, \tau_s, \tau_e) \in \mathcal{F} \;\big|\; r = \hat{r}_i \,\wedge\, \pi_{\bar{\rho}}(f) = \hat{e}_i \,\big\}. \tag{8}$$

The role restriction $\pi_{\bar{\rho}}(f) = \hat{e}_i$ is Definition 1 in symbolic form, discarding facts where the anchor plays the wrong role; the union $\mathcal{S} = \bigcup_i \mathrm{Query}(\hat{\psi}_i, \rho_i)$ is the *structurally consistent evidence set*.

Single-event temporal filtering. For a window $\Omega = [\tau_s^q, \tau_e^q] \subseteq \mathcal{T}^\star$, the granularity-aware filter keeps only those facts of $\mathcal{S}$ whose interval lies inside Ω under the truncation $\lfloor\cdot\rfloor_\gamma$,

$$\mathrm{Filter}_\gamma(\mathcal{S}, \Omega) \;=\; \big\{\, f \in \mathcal{S} \;\big|\; \lfloor\tau_s^q\rfloor_\gamma \preceq \lfloor\tau_s^{(f)}\rfloor_\gamma \;\wedge\; \lfloor\tau_e^{(f)}\rfloor_\gamma \preceq \lfloor\tau_e^q\rfloor_\gamma \,\big\}. \tag{9}$$

Because every comparison uses $\preceq$ on $\mathcal{T}^\star$, this one formula realizes the equal, before, and after constraints through the degenerate windows $[\tau, \tau]$, $[\tau_{-\infty}, \tau)$, and $(\tau, \tau_{+\infty}]$ with no case distinction for sentinels, and setting $\mathcal{S} \leftarrow \mathrm{Filter}_\gamma(\mathcal{S}, \Omega)$ enforces the window condition of Definition 2.

Inter-event temporal alignment. When $k \geq 2$, an alignment operator narrows $\mathcal{S}$ by enforcing the relations in Λ. With $\mathcal{S}_a \subseteq \mathcal{S}$ the anchor set, $\mathcal{S}_r$ the reference set of a relation $\lambda \in \mathcal{L}$, and envelope endpoints $\tau_s^r = \min_{g \in \mathcal{S}_r} \tau_s^{(g)}$ and $\tau_e^r = \max_{g \in \mathcal{S}_r} \tau_e^{(g)}$, the operator is

$$\mathrm{Align}_\lambda(\mathcal{S}_a, \mathcal{S}_r) \;=\; \big\{\, f \in \mathcal{S}_a \;\big|\; \mathrm{rel}_\lambda(\tau_s^{(f)}, \tau_e^{(f)}, \tau_s^r, \tau_e^r) \,\big\}, \tag{10}$$

where rel_λ is the Boolean predicate induced by λ. All six predicates are enumerated in Appendix B.6.

Theorem 1 (Consistency of the symbolic operators)**.** *(i) Every fact returned by* $\mathrm{Query}(\hat{\psi}_i, \rho)$ *in Eq.* (8) *is structurally consistent with* $\hat{\psi}_i$*. (ii) Every fact surviving* Filter_γ *in Eq.* (9) *and, when* $k \geq 2$*,* Align_λ *in Eq.* (10) *is temporally consistent with* $\hat{\mathcal{C}}_q$*.*

Multi-event composition. When α is a duration, ranked position, or set-operation result rather than an entity or boundary timestamp, a composition operator applies,

$$\mathrm{Compose}_\theta(f_1, \ldots, f_k) \;=\; \Theta_\theta\Big(\tau_s^{(1)}, \tau_e^{(1)}, \ldots, \tau_s^{(k)}, \tau_e^{(k)}\Big), \tag{11}$$

where Θ_θ is the arithmetic function selected by θ. Three representative instances are: aggregated duration returns $\sum_i (\tau_e^{(i)} - \tau_s^{(i)})$, interval intersection returns $[\max_i \tau_s^{(i)}, \min_i \tau_e^{(i)}]$, and ranking sorts by a key from $\{\tau_s, \tau_e, \tau_e - \tau_s\}$ and returns the position of the target fact.

Evidence set and candidate list. The three operators compose into a faithful evidence set,

$$\mathcal{F}_q \;=\; \mathrm{Align}_\Lambda\Big(\mathrm{Filter}_\gamma\big(\textstyle\bigcup_{i=1}^k \mathrm{Query}(\hat{\psi}_i,\, \rho_i),\; \Omega\big),\; \mathcal{S}_{\mathrm{ref}}\Big)\Big|_{\leq \beta}, \tag{12}$$

where $\mathcal{S}_{\mathrm{ref}} = \bigcup_{j \neq a_q} \mathrm{Query}(\hat{\psi}_j, \rho_j)$ collects reference facts from the templates other than the anchor a_q and $|_{\leq \beta}$ caps the surviving set at β facts while keeping every fact at the temporal boundary; we set $\beta = 30$ throughout. Filter_γ is the identity when $\Omega = \bot$, and $\mathrm{Align}_\Lambda = \bigcap_{\lambda \in \Lambda} \mathrm{Align}_\lambda$ is the identity when $k = 1$. Query secures Definition 1 and $\mathrm{Filter}_\gamma, \mathrm{Align}_\Lambda$ secure Definition 2, the tie-preserving cap β stays inside the exception Definition 3 allows, and a post-execution check re-verifies every fact.

Theorem 2 (Faithfulness)**.** *Under Assumption 1, the evidence set* $\mathcal{F}_q$ *of Eq.* (12) *is faithful to* $\hat{\mathcal{C}}_q$*.*

The candidate list Cand is extracted from $\mathcal{F}_q$ by dispatching on the answer slot α, writing $\mathcal{A}_{\mathrm{bnd}} = \{\mathsf{subj}, \mathsf{obj}, \mathsf{tss}, \mathsf{tse}, \mathsf{tsr}\}$ for the boundary slots and $\mathcal{A}_{\mathrm{cmp}} = \{\mathsf{dur}, \mathsf{rdur}, \mathsf{rank}, \mathsf{setop}\}$ for the arithmetic slots, so that the dispatch reads

$$\mathrm{Cand} \;=\; \begin{cases} \mathrm{Boundary}_\gamma(\mathcal{F}_q, \alpha, \theta), & \alpha \in \mathcal{A}_{\mathrm{bnd}}, \\ \{\mathrm{Compose}_\theta(f_1, \ldots, f_k)\}, & \alpha \in \mathcal{A}_{\mathrm{cmp}}, \end{cases} \tag{13}$$

where the composition branch ranges over every tuple of facts drawn one per template from $\mathcal{F}_q$, and $\mathrm{Boundary}_\gamma$ sorts $\mathcal{F}_q$ by the temporal key of (α, θ) and returns every entity or timestamp at the boundary. For arithmetic slots $\mathrm{Compose}_\theta$ usually gives a unique output, so $|\mathrm{Cand}| = 1$.

4.4 Stage 4: Adaptive Reasoning

Stage 4 determines $\hat{a}$ from $\hat{\mathcal{C}}_q$, $\mathcal{F}_q$, and Cand, skipping the second LLM call when Cand is a singleton or θ carries an ordering directive, where top takes the first element of $\mathrm{Boundary}_\gamma$'s order. Otherwise the LLM sees only $\mathcal{F}_q$, so its answer is drawn from an evidence set that Theorem 2 certifies, and the budget B bounds how often a hallucination can enter the pipeline.

Deterministic shortcut. Let $\mathsf{SortDirected}(\hat{\mathcal{C}}_q)$ hold when θ carries an ordering directive such as "first" or "last". A question admits a *deterministic shortcut* when

$$\mathsf{Shortcut}(\hat{\mathcal{C}}_q, \mathrm{Cand}) \;\equiv\; \big(\,|\mathrm{Cand}| = 1\,\big) \;\vee\; \mathsf{SortDirected}(\hat{\mathcal{C}}_q). \tag{14}$$

The first disjunct covers every case in which Stage 3 narrows Cand to one element, which holds for most duration, ranking, and set-operation questions; the second covers the "first" and "last" questions that the boundary extractor already resolves.

Adaptive answer rule. Whether the second LLM call of FaiTH is issued depends on Eq. (14):

$$(\hat{a},\, B') \;=\; \begin{cases} \big(\mathrm{top}(\mathrm{Cand}),\; B\big), & \text{if } \mathsf{Shortcut}, \\ \big(\textsc{Reason}(q, \mathcal{F}_q, \hat{\mathcal{C}}_q),\; B{-}1\big), & \text{otherwise}, \end{cases} \tag{15}$$

where REASON receives q, $\hat{\mathcal{C}}_q$, and a chronological rendering of $\mathcal{F}_q$, and may answer only with entities or timestamps in $\mathcal{F}_q$. The compile call and the optional reason call are the only two call sites, so FaiTH never issues more than two LLM calls per question. On an empty evidence set a symbolic recovery subroutine rewrites $\hat{\mathcal{C}}_q$ deterministically and re-executes Stage 3 without consuming budget (Appendix B.5); Algorithm 1 in Appendix B.7 gives the pseudo-code.

Cost of one question. Write c_{LLM} for the cost of one LLM call and $d_{\max}$ for the largest number of facts incident to any entity. Stages 1 and 4 contribute at most $2c_{\mathrm{LLM}}$, Stage 2 costs $O(1)$ per mention after a one-off index build shared across questions, and Stage 3 retrieves at most $d_{\max}$ facts per template and sorts them, so one question costs $O(c_{\mathrm{LLM}} + k \cdot d_{\max} \log d_{\max})$, dominated in practice by the one or two LLM calls (Appendix B.8).

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate on MULTITQ (Chen et al., 2023), TIMELINEKGQA-ICEWS (TL-ICEWS), and TIMELINEKGQA-CRON (TL-CRON) (Boschee et al., 2015; Saxena et al., 2021; Sun et al., 2025). We probe robustness with three cloud-hosted backbones, GPT-5.4 (OpenAI, 2026), DeepSeek-V3.2 (DS-V3.2) (Liu et al., 2025), and DeepSeek-V4-Flash (DS-V4-Flash) (Xu et al., 2026), and with eight local deployments of open-weight Gemma 4 (Team et al., 2026), Qwen 3.6 (Yang et al., 2025), and GLM-4.7-Flash (Zeng et al., 2025) models served by Ollama (Ollama, 2026) on an Apple Mac Studio with an M4 Max chip and 36 gigabytes (GB) of unified memory. Every LLM runs in non-thinking mode at temperature 0, and both calls share one backbone. Following Gong et al. (2025) we report Hits@1, plus Hits@10 on MULTITQ; Appendices C.1 and C.2 cover benchmarks and metrics.

5.2 MAIN RESULTS

Performance on MULTITQ. Table 1 compares FaiTH against published baselines on MULTITQ. FaiTH (GPT-5.4) reaches an overall Hits@1 of 0.852 against 0.780 for Temp-R1 (Gong et al., 2026) and 0.779 for PoK (Qian et al., 2025), and the margin is widest on multiple-constraint questions, where FaiTH scores 0.722 with GPT-5.4 and 0.716 with DS-V3.2 against 0.550 for Temp-R1 and 0.424 for RTQA (Gong et al., 2025). The symbolic engine explains that gap: Stage 3 settles most multi-constraint questions before any LLM reasoning. FaiTH (DS-V4-Flash) reaches 0.817 and the local 20 GB 4-bit Gemma4:31b-nvfp4 reaches 0.805, beating every published baseline overall, though four baselines score higher on time answers.

Performance on TIMELINEKGQA. Table 2 covers TimelineKGQA. On TL-CRON, FaiTH (DS-V4-Flash) reaches 0.7188 overall with the best Medium and Complex scores; on TL-ICEWS, FaiTH (DS-V3.2) leads on Medium while the local FaiTH (Gemma4:31b-nvfp4) takes the best overall and Complex scores, 0.6971 and 0.7122, so the backbone matters only secondarily there.

LLM call budget and API cost. Table 7 in Appendix C.3 reports per-question call counts and API costs. Across all three benchmarks, with DS-V3.2 and GPT-5.4, the deterministic shortcut settles

Table 1: Hits@1 and Hits@10 by question type and answer type on the MULTITQ test set. Best and second-best per column are **bold** and *italic*; baselines are quoted from Gong et al. (2025) and Gong et al. (2026). Hits@10 is unavailable for baselines whose source paper does not report it. All 11 configurations appear in Section 5.3 and Appendix C.5.

	Hits@1					Hits@10				
		Question Type		Answer Type			Question Type		Answer Type	
Model	Overall	Multiple	Single	Entity	Time	Overall	Multiple	Single	Entity	Time
BERT (Reimers & Gurevych, 2019)	0.083	0.061	0.092	0.101	0.040	0.441	0.392	0.461	0.531	0.222
DistillBERT (Sanh et al., 2019)	0.083	0.074	0.087	0.102	0.037	0.482	0.426	0.505	0.591	0.216
ALBERT (Lan et al., 2019)	0.108	0.086	0.116	0.139	0.032	0.484	0.415	0.512	0.589	0.228
LLaMA2-7B (Touvron et al., 2023)	0.185	0.101	0.220	0.239	0.055	–	–	–	–	–
ChatGPT (OpenAI, 2023)	0.102	0.077	0.147	0.137	0.020	–	–	–	–	–
EmbedKGQA (Saxena et al., 2020)	0.206	0.134	0.235	0.290	0.001	0.459	0.439	0.467	0.648	0.001
CronKGQA (Saxena et al., 2021)	0.279	0.134	0.337	0.328	0.156	0.608	0.453	0.671	0.696	0.392
MultiQA (Chen et al., 2023)	0.293	0.159	0.347	0.349	0.157	0.635	0.519	0.682	0.733	0.396
ARI (Chen et al., 2024b)	0.380	0.210	0.680	0.394	0.344	–	–	–	–	–
TimeR^4 (Qian et al., 2024)	0.728	0.335	0.887	0.639	0.945	–	–	–	–	–
RTQA (Gong et al., 2025)	0.765	0.424	0.902	0.692	0.942	0.768	0.427	*0.907*	0.697	**0.942**
PoK (Qian et al., 2025)	0.779	0.409	**0.929**	0.696	*0.962*	–	–	–	–	–
Temp-R1 (Gong et al., 2026)	0.780	0.550	0.888	0.714	**0.969**	–	–	–	–	–
FaiTH (GPT-5.4)	**0.852**	**0.722**	*0.905*	**0.830**	0.908	**0.880**	**0.773**	**0.923**	**0.868**	*0.908*
FaiTH (DS-V3.2)	*0.826*	*0.716*	0.870	*0.817*	0.847	0.859	*0.768*	0.895	*0.863*	0.848
FaiTH (DS-V4-Flash)	0.817	0.699	0.865	0.795	0.870	*0.865*	0.760	*0.907*	0.862	0.870
FaiTH (Gemma4:31b-nvfp4)	0.805	0.709	0.844	0.784	0.857	0.854	0.761	0.892	0.852	0.857

Table 2: Hits@1 by question difficulty on the two TIMELINEKGQA benchmarks. Best and second-best per column are **bold** and *italic*; baselines are quoted from Gong et al. (2026). FaiTH rows carry four decimals, baselines three. All 11 configurations appear in Section 5.3 and Appendix C.5.

	TIMELINEKGQA-CRON				TIMELINEKGQA-ICEWS			
Model	Overall	Simple	Medium	Complex	Overall	Simple	Medium	Complex
RAG Baseline (Lewis et al., 2020)	0.235	0.704	0.092	0.009	0.265	0.660	0.128	0.011
LLaMA2-7B (Touvron et al., 2023)	0.169	0.049	0.143	0.282	0.111	0.035	0.066	0.322
GPT-4o (OpenAI, 2024a)	0.206	0.069	0.130	0.376	0.113	0.051	0.035	0.353
RTQA (Gong et al., 2025)	0.298	0.608	0.218	0.135	–	–	–	–
PoK (Qian et al., 2025)	0.651	0.737	0.539	0.683	0.602	0.744	0.456	0.578
Temp-R1 (Gong et al., 2026)	0.705	**0.960**	0.486	0.672	0.642	**0.866**	0.388	0.595
FaiTH (GPT-5.4)	0.6847	0.8321	0.5562	0.6846	0.6864	0.8325	0.5311	0.6718
FaiTH (DS-V3.2)	*0.7067*	0.8254	*0.5780*	*0.7285*	*0.6955*	0.8340	**0.5388**	0.6970
FaiTH (DS-V4-Flash)	**0.7188**	0.8292	**0.5903**	**0.7466**	0.6943	0.8298	0.5328	*0.7088*
FaiTH (Gemma4:31b-nvfp4)	0.6822	0.8283	0.5467	0.6890	**0.6971**	*0.8353*	0.5325	**0.7122**
FaiTH (Qwen3.6:27b-nvfp4)	0.6982	*0.8325*	0.5569	0.7191	0.6878	0.8332	0.5276	0.6824
FaiTH (GLM-4.7-Flash:q4_K_M)	0.7023	0.8288	0.5667	0.7241	0.6853	0.8160	*0.5341*	0.6921

64.66% to 71.62% of questions with the single compile call, and FaiTH (DS-V3.2) processes all three test sets for under $30, whereas RTQA (Gong et al., 2025) averages 3.96 calls per question and Temp-R1 (Gong et al., 2026) adds a one-time training cost near $50 to generate fine-tuning samples from GPT-4o (OpenAI, 2024a). Appendix C.10 breaks the shortcut rate down by question type.

5.3 ROBUSTNESS ACROSS LLM BACKBONES

We evaluate FaiTH under 11 LLM backbone configurations spanning 9 base models from 5 families: the three cloud-hosted models of Section 5.1 and eight open-weight deployments served locally with Ollama. GLM-4.7-Flash and Qwen3.6:27b each appear at two quantizations, so Table 3 isolates precision at a fixed base model as well as the base model itself, and every configuration shares one set of prompts, symbolic engine, and evaluation scripts. The spread is narrow on the TimelineKGQA benchmarks, where ten of the 11 configurations on TL-ICEWS sit within four points of the best score, and wider on MULTITQ, whose day-level granularity and larger answer vocabulary make

Table 3: Overall Hits@1 (%) of FaiTH under 11 LLM backbone configurations spanning 9 base models. Local backbones run on one Apple Mac Studio with M4 Max and 36 GB unified memory, and Size is the on-disk footprint of the quantized weights. Best and second-best FaiTH scores per benchmark are **bold** and *italic*; Temp-R1 is listed for reference and is not ranked.

Backbone	Deploy	Size	MULTITQ	TL-CRON	TL-ICEWS
GPT-5.4	Cloud		**85.20**	68.47	68.64
DS-V3.2	Cloud		*82.60*	*70.67*	*69.55*
DS-V4-Flash	Cloud		81.70	**71.88**	69.43
Gemma4:31b-nvfp4	Local	20 GB	80.52	68.22	**69.71**
Qwen3.6:27b-mtp-q8_0	Local	29 GB	78.08	69.64	69.24
Gemma4:12b-mlx-bf16	Local	23 GB	77.18	65.53	61.54
Gemma4:26b-mxfp8	Local	26 GB	73.12	62.44	66.28
Qwen3.6:27b-nvfp4	Local	19 GB	69.81	69.82	68.78
GLM-4.7-Flash:q4_K_M	Local	19 GB	69.29	70.23	68.53
GLM-4.7-Flash:q8_0	Local	31 GB	67.70	69.25	68.20
Qwen3.6:35b-a3b-nvfp4	Local	21 GB	64.37	66.89	67.56
Temp-R1 (SOTA baseline)			78.00	70.50	64.20

Table 4: Ablations (Hits@1) of FaiTH (DS-V3.2). *w/o ra* drops the role restriction of Eq. (8); *w/o tf* drops the window of Eq. (9) and the alignment of Eq. (10); *w/o ds* disables Eqs. (14) and (15), forcing one reason call per question. Best and second-best per column are **bold** and *italic*.

	MULTITQ					TIMELINEKGQA-ICEWS			
Variant	Overall	Single	Multiple	Entity	Time	Overall	Simple	Medium	Complex
FaiTH, full	**0.826**	*0.870*	**0.716**	**0.817**	0.847	**0.696**	**0.834**	**0.539**	**0.697**
w/o ra	0.763	0.834	*0.587*	0.747	0.801	0.548	0.612	0.445	0.599
w/o tf	0.508	0.651	0.152	0.342	**0.911**	*0.681*	*0.810*	*0.531*	*0.692*
w/o ds	*0.789*	**0.872**	0.586	*0.763*	*0.854*	0.456	0.757	0.224	0.285

Stage-1 compilation harder for small models. On the TimelineKGQA benchmarks quantized local models close the gap: Gemma4:31b-nvfp4 at 20 GB leads TL-ICEWS with 69.71%, ahead of DS-V3.2 and GPT-5.4, and GLM-4.7-Flash:q4_K_M at 19 GB reaches 70.23% on TL-CRON against 70.67% for DS-V3.2. On MULTITQ no local model reaches the weakest cloud backbone.

The two quantization pairs behave differently. GLM-4.7-Flash at q4_K_M (19 GB) matches or beats its q8_0 counterpart (31 GB) on all three benchmarks, whereas Qwen3.6:27b gains 8.27 points on MULTITQ between nvfp4 (19 GB) and mtp-q8_0 (29 GB) yet stays within half a point on both TimelineKGQA benchmarks. Precision therefore matters only where Stage-1 compilation is hardest, and not where the symbolic engine already fixes the answer. Ten of the 11 configurations beat Temp-R1 on TL-ICEWS, five on MULTITQ, and two on TL-CRON (Appendix C.5).

5.4 ABLATION STUDIES

Table 4 isolates three design components. Removing role-aware retrieval (*w/o ra*) drops overall Hits@1 from 0.826 to 0.763 on MULTITQ and from 0.696 to 0.548 on TL-ICEWS, confirming Definition 1. Removing temporal filtering (*w/o tf*) collapses MULTITQ multiple-constraint questions ($0.716 \rightarrow 0.152$) and entity answers ($0.817 \rightarrow 0.342$); the rise in the Time score ($0.847 \rightarrow 0.911$) traces to Stage-1 compilation errors on some *first_last* questions, a backbone-specific artifact absent with DS-V4-Flash (Appendices C.4 and C.7). Disabling the shortcut (*w/o ds*) lowers overall Hits@1 to 0.789 on MULTITQ and 0.456 on TL-ICEWS, since Stage 3 already settles the ordering-directed and arithmetic questions and routing them through the LLM adds hallucination risk.

5.5 SENSITIVITY ANALYSES

Figure 2 scans the two hyperparameters a user of FaiTH must set. The top row varies the adaptive bound β of Eq. (12): overall Hits@1 stays flat as β grows from 2 to 50 on both benchmarks, and so

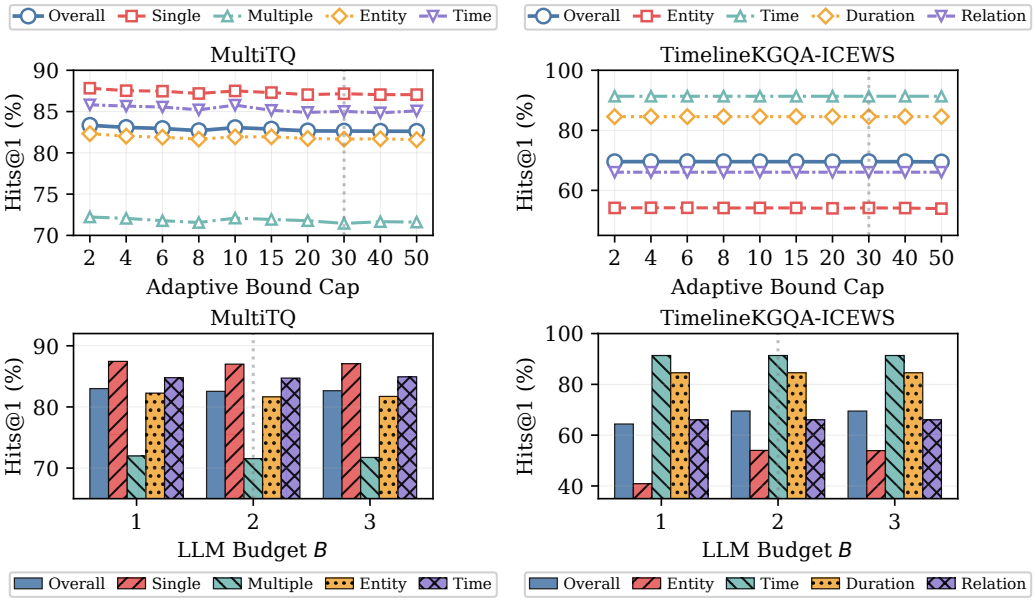

Figure 2: Sensitivity of FaiTH (DS-V3.2) on MULTITQ (left) and TL-ICEWS (right). The top row scans the adaptive bound β of Eq. (12) and the bottom row the budget B, where $B = 3$ permits an extra reason call after empty-evidence recovery, outside the default two-call design. The TL-ICEWS panels group the nine answer slots into four size-weighted buckets (Appendix C.2), and dotted lines mark the default ($\beta = 30$, $B = 2$). Each panel is an independent sweep, so its default point can differ by a few tenths from Tables 1 and 2.

does every answer-type bucket. The bound never removes a fact tied at the temporal boundary, and boundary facts carry the answer, so a small cap discards little and Theorem 2 holds for every β.

The bottom row scans the per-question budget B. At $B = 1$ the compile call is the only LLM call and the engine answers from Cand, reaching 83.00% on MULTITQ but only 64.38% on TL-ICEWS. The five points or so that the reason call adds on TL-ICEWS come from the Entity bucket alone, where several entities often survive Stage 3, while the Time, Duration, and Relation buckets do not move because Stage 3 already fixes them. On MULTITQ the same call costs under half a point, so the top-ranked candidate is as reliable there as the LLM. Raising the budget to $B = 3$ moves no score by more than 0.2 points, so the two-call ceiling costs nothing rather than binding. The appendix adds TKG-noise robustness (C.6), finer ablations (C.7), answer coverage (C.8), error attribution (C.9), shortcut statistics (C.10), further metrics (C.11), and automated grounding (C.12).

6 CONCLUSION

We present FaiTH, a training-free framework that addresses structural and temporal unfaithfulness in TKGQA through a four-stage *compile-ground-execute-reason* pipeline. It produces provably faithful evidence sets and reaches state-of-the-art Hits@1 on three heterogeneous benchmarks at low API cost, and across 11 backbone configurations spanning 9 base models the TimelineKGQA spread stays narrow once the evidence set is faithful, where quantized 19 to 20 GB models rival cloud-hosted LLMs. A natural next step is to extend the faithfulness definitions to spatial and numeric constraints.

Limitations. The guarantees rest on Assumption 1, so a mention the cascade fails to resolve makes the engine execute a wrong specification. Stage-1 compilation is a second error source that nothing downstream can detect once a specification is well formed but wrong, and the cap of three event templates limits generalization. Finally, cloud backbones are not bit-reproducible even at temperature 0: four full runs of FaiTH (DS-V4-Flash) on MULTITQ span 0.51 points (Appendix C.2).

AI USE STATEMENT

In this work, we used Claude Opus 5 (Anthropic, 2026b) and Claude Fable 5.1 (Anthropic, 2026a) as generative AI tools to correct grammar and improve readability. We have not used generative AI tools to generate data, propose hypotheses, design the method or implement the experiments, or supply any step of a proof; translation, dataset cleaning, and qualitative analysis are not applicable to this work. Additionally, we used generative AI tools to interpret results, namely to draft the discussion of the backbone comparison (Section 5.3) and of the sensitivity analyses (Section 5.5). We have reviewed all AI-assisted work: every modified paragraph was checked against the ideas or results it describes, and every reported number was traced back to the experiment logs. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

FaiTH is evaluated only on publicly available benchmarks distributed under their own licenses, involves no human subjects and no annotation, and collects no personal data beyond the public entity names already present in those benchmarks. Two of the three benchmarks derive from the ICEWS event data, which record interactions between political actors as reported in news media, so a system built on FaiTH could in principle be applied to politically sensitive questions; we encourage practitioners deploying temporal knowledge graph reasoning in such settings to add appropriate safeguards. The faithfulness guarantees of Section 4.3 hold under Assumption 1 and do not certify the Stage-1 compilation, which an LLM produces. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

FaiTH is training-free, so reproducing it requires no model training. Section 5.1 lists the benchmarks, backbones, decoding settings, and hardware, Appendix C.1 describes the benchmarks and baselines, Appendix C.2 gives the evaluation protocol and the run-to-run variance of the cloud backbones, and Appendix B.3 describes the per-benchmark instantiation of the grounding cascade and the compile prompt. Every assumption behind the theoretical claims is stated in Section 4.3, and the complete proofs are in Appendix B.6. Algorithm 1 in Appendix B.7 gives the full pipeline, and Appendix D shows the Stage-1 compile and Stage-4 reason prompt templates for MULTITQ and TL-ICEWS with the few-shot examples abridged; the repository contains every prompt verbatim. The anonymous repository at `https://anonymous.4open.science/r/FaiTH-978E/` contains detailed implementations of the symbolic constraint execution engine, the grounding cascade, the structural compilation and reasoning prompts, and the evaluation scripts.

REFERENCES

James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26 (11):832–843, 1983.

Anthropic. Claude: Fable 5.1 and mythos 5.1, 2026a. URL `https://www.anthropic.com/claude-fable-and-mythos-5-1`. Accessed: 2026-09-22.

Anthropic. Introducing claude opus 5, 2026b. URL `https://www.anthropic.com/news/claude-opus-5`. Accessed: 2026-09-22.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. ICEWS Coded Event Data, 2015. URL `https://doi.org/10.7910/DVN/28075`.

Zhuo Chen, Zhao Zhang, Zixuan Li, Fei Wang, Yutao Zeng, Xiaolong Jin, and Yongjun Xu. Self-improvement programming for temporal knowledge graph question answering. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 14579–14594, Torino, Italia,

May 2024a. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1270/.

Ziyang Chen, Jinzhi Liao, and Xiang Zhao. Multi-granularity temporal question answering over knowledge graphs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11378–11392, 2023.

Ziyang Chen, Dongfang Li, Xiang Zhao, Baotian Hu, and Min Zhang. Temporal knowledge question answering via abstract reasoning induction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4872–4889, 2024b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Zhaoyan Gong, Juan Li, Zhiqiang Liu, Lei Liang, Huajun Chen, and Wen Zhang. Rtqa: Recursive thinking for complex temporal knowledge graph question answering with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9864–9881, 2025.

Zhaoyan Gong, Zhiqiang Liu, Songze Li, Xiaoke Guo, Yuanxiang Liu, Xinle Deng, Zhizhen Liu, Lei Liang, Huajun Chen, and Wen Zhang. Temp-r1: A unified autonomous agent for complex temporal kgqa via reverse curriculum reinforcement learning. *arXiv preprint arXiv:2601.18296*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, pp. 1057–1062, 2018.

Zhen Jia, Soumajit Pramanik, Rishiraj Saha Roy, and Gerhard Weikum. Complex temporal question answering on knowledge graphs. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 792–802, 2021.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Yonghao Liu, Di Liang, Fang Fang, Sirui Wang, Wei Wu, and Rui Jiang. Time-aware multiway adaptive fusion network for temporal knowledge graph question answering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Yonghao Liu, Di Liang, Mengyu Li, Fausto Giunchiglia, Ximing Li, Sirui Wang, Wei Wu, Lan Huang, Xiaoyue Feng, and Renchu Guan. Local and global: Temporal question answering via information fusion. In *IJCAI*, pp. 5141–5149, 2023b.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models, 2025. URL https://arxiv.org/abs/2410.13080.

Costas Mavromatis, Prasanna Lakkur Subramanyam, Vassilis N Ioannidis, Adesoji Adeshina, Phillip R Howard, Tetiana Grinberg, Nagib Hakim, and George Karypis. Tempoqr: temporal question reasoning over knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 5825–5833, 2022.

Ollama. The easiest way to build with open models, 2026. URL https://ollama.com/. Accessed: 2026-08-01.

OpenAI. Gpt-3.5 turbo model | openai api. https://developers.openai.com/api/docs/models/gpt-3.5-turbo, 2023. Accessed: 2026-03-24.

OpenAI. Hello GPT-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/. Accessed: 2026-03-24.

OpenAI. GPT-4o mini: Advancing cost-efficient intelligence, 2024b. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2026-03-24.

OpenAI. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026. Accessed: 2026-04-14.

OpenAI. Introducing gpt-6 sol and luna, 2026. URL https://openai.com/index/introducing-gpt-6-sol-and-luna/. Accessed: 2026-09-24.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, Li Zhang, and Kehui Song. Timer4: Time-aware retrieval-augmented large language models for temporal knowledge graph question answering. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 6942–6952, 2024.

Xinying Qian, Ying Zhang, Yu Zhao, Baohang Zhou, Xuhui Sui, and Xiaojie Yuan. Plan of knowledge: Retrieval-augmented large language models for temporal knowledge graph question answering. *arXiv preprint arXiv:2511.04072*, 2025.

QianyiHu QianyiHu, Xinhui Tu, Guo Cong, and Shunping Zhang. Time-aware react agent for temporal knowledge graph question answering. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6013–6024, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S. Zimmermann, and Wieland Brendel. Infonce: Identifying the gap between theory and practice, 2025. URL https://arxiv.org/abs/2407.00143.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4498–4507, 2020.

Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6663–6676, 2021.

Aditya Sharma, Apoorv Saxena, Chitrank Gupta, Mehran Kazemi, Partha Talukdar, and Soumen Chakrabarti. Twirgcn: Temporally weighted graph convolution for question answering over temporal knowledge graphs. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2049–2060, 2023.

Miao Su, Zixuan Li, Zhuo Chen, Long Bai, Xiaolong Jin, and Jiafeng Guo. Temporal knowledge graph question answering: A survey. *arXiv preprint arXiv:2406.14191*, 2024.

Qiang Sun, Sirui Li, Du Huynh, Mark Reynolds, and Wei Liu. Timelinekgqa: A comprehensive question-answer pair generator for temporal knowledge graphs. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 797–800, 2025.

Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. Memotime: Memory-augmented temporal knowledge graph enhanced large language model reasoning. *arXiv preprint arXiv:2510.13614*, 2025.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

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

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pp. 2071–2080. PMLR, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL `https://arxiv.org/abs/1706.03762`.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2717–2739, 2023a.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 2609–2634, 2023b.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Zhiyuan Zha, Pengnian Qi, Xigang Bao, Mengyuan Tian, and Biao Qin. M 3 tqa: Multi-view, multi-hop and multi-stage reasoning for temporal question answering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 10086–10090. IEEE, 2024.

A Extended Related Work

A.1 Embedding and GNN Methods

Early TKGQA methods cast the problem as TKG completion, scoring candidate answers through embedding-based similarity. CronKGQA (Saxena et al., 2021) combines TComplEx (Lacroix et al., 2020) temporal KG embeddings with BERT-based (Devlin et al., 2019) question representations, achieving strong results on simple questions but degrading on complex temporal reasoning. EmbedKGQA (Saxena et al., 2020) applies static ComplEx (Trouillon et al., 2016) embeddings to temporal settings, lacking explicit temporal modeling. TempoQR (Mavromatis et al., 2022) introduces time-scope features and entity-as-expert representations to enrich question encoding. MultiQA (Chen et al., 2023) extends CronKGQA with a multi-granularity aggregation module that fuses day-level embeddings into coarser ones with a transformer encoder (Vaswani et al., 2023).

Graph neural networks (GNNs) and GNN-based approaches model structural dependencies explicitly. EXAQT (Jia et al., 2021) combines relational graph convolutional networks with dictionary-based entity matching. TwiRGCN (Sharma et al., 2023) introduces temporally weighted graph convolutions with answer gating. LGQA (Liu et al., 2023b) employs multi-hop message-passing GNNs to jointly encode local and global temporal contexts, while TMA (Liu et al., 2023a) proposes a time-aware multi-path adaptive fusion network. M3TQA (Zha et al., 2024) enables asynchronous alignment of pre-trained language model and GNN features through multi-stage aggregation. Despite their structural advantages, these methods struggle with complex multi-hop temporal questions and remain limited to the answer vocabularies defined by large-scale TKG database.

A.2 LLM-based Methods

Recent work leverages large language models (LLMs) to formulate question answering as a generative task. By appropriately integrating TKGQA datasets with their corresponding TKG databases, a more informative prompt could be engineered for each question so that LLMs could have a higher likelihood of generating a correct answer. There are three major paradigms for LLM-based approaches.

Prompt-engineered pipelines. These methods use closed-source LLMs as reasoners within fixed, multi-stage workflows. ARI (Chen et al., 2024b) augments question understanding through adaptive reasoning instructions. TempAgent (QianyiHu et al., 2025) adapts the ReAct (Yao et al., 2022) framework with a temporal reasoning toolkit. MemoTime (Tan et al., 2025) stores and retrieves solution paths of previously solved questions as external memory. RTQA (Gong et al., 2025) decomposes complex temporal questions into sub-problems solved recursively. While effective, these approaches require multiple API calls per question at inference time (e.g., RTQA averages 3.96 calls per question (Gong et al., 2025)), and none provides a mechanism to verify whether the evidence presented to the LLM is structurally and temporally faithful to the underlying TKG.

Retrieval augmentation. These methods combine retrievers with fine-tuned open-source LLMs. TimeR^4 (Qian et al., 2024) performs question rewriting to expose implicit temporal constraints, using a fine-tuned Sentence-BERT (Reimers & Gurevych, 2019) retriever and Llama-2-7B (Touvron et al., 2023) generator, though it still requires GPT-3.5 Turbo (OpenAI, 2023) for the rewriting stage. PoK (Qian et al., 2025) extends this with a plan of knowledge framework that decomposes questions into structured sub-objectives by prompting GPT-4o (OpenAI, 2024a), paired with a contrastive time-aware retriever fine-tuned with an InfoNCE objective (Rusak et al., 2025).

RL-trained autonomous agents. Reinforcement learning (RL) is also involved in TKGQA. Temp-R1 (Gong et al., 2026) represents a paradigm shift, formulating TKGQA as a Markov decision process and fine-tuning Llama-3.1-8B (Grattafiori et al., 2024) with group relative policy optimization (GRPO) strategy (Guo et al., 2025), which expands the agent's action space with explicit `<plan>`, `<filter>`, and `<rank>` operations, and introduces reverse curriculum learning to prevent shortcut behavior. Search-R1 (Jin et al., 2025) applies similar RL principles to teach models when and how to invoke external search tools. While Temp-R1 achieves state-of-the-art results without prompting commercialized LLMs during inference, it requires a sophisticated multi-stage training pipeline including supervised fine-tuning cold start data generated by closed-source GPT-4o.

A.3 Symbolic Execution in KGQA

Beyond the temporal setting, faithfulness of LLM reasoning over KGs has been pursued through constrained decoding and graph-guided search, for example graph-constrained reasoning (Luo et al., 2025) and self-verification mechanisms (Chen et al., 2024a). Semantic-parsing-style KGQA compiles questions into executable logical forms over static KGs, an idea that inspires our Stage-1 compilation. However, static logical forms lack native support for interval arithmetic, granularity truncation, sentinel endpoints, and Allen-style inter-event relations, which are exactly the constructs formalized by the FaiTH specification and executed by the Stage-3 operators. To the best of our knowledge, FaiTH is the first TKGQA framework that couples a dataset-agnostic temporal specification with a symbolic executor whose output evidence is provably structurally and temporally consistent.

B Extended Methodological Details

B.1 Key Challenges and Design Motivations

This appendix elaborates the two challenges summarized in Section 1 and connects each design decision of FaiTH to the failure mode it addresses.

Challenge 1: guaranteeing evidence faithfulness. In the *retrieve-then-reason* paradigm, the evidence set handed to the LLM is assembled by approximate similarity search. Two distinct kinds of inconsistency arise. *Structural* inconsistency occurs when a retrieved fact matches the topical content of the question but assigns the anchor entity to the wrong semantic role, for example returning $(x, \textit{Investigate}, e)$ for a question about whom e investigated. *Temporal* inconsistency occurs when a retrieved fact lies outside the time window of the question, or violates the ordering relation between events in a compositional question, for example returning a 2007 fact for a question about events before June 2006. Dense retrievers encode both roles and timestamps only distributionally, so neither property can be enforced as a hard condition, and reranking merely reorders a candidate pool that may not contain a fully consistent fact at all. The consequence is quantified by the answer coverage of existing pipelines. PoK reports 71.78% top-50 coverage on MultiTQ, so on more than a quarter of the questions the reasoner has to answer without a retrieved fact that carries the gold answer. FaiTH's response to this challenge is to replace approximate retrieval with symbolic constraint execution over a compiled specification, so that structural consistency is enforced by construction through role-aware retrieval (Eq. (8)) and temporal consistency through granularity-aware filtering and alignment (Eqs. (9) and (10)), yielding the faithfulness guarantee following Definition 3.

Challenge 2: bounding unfaithfulness amplification. When the evidence is unreliable, existing systems compensate by invoking the LLM repeatedly, through iterative retrieval refinement, tree-structured decomposition, or agentic tool loops. Each additional call is an independent opportunity for hallucination, and because later calls condition on earlier outputs, errors compound rather than cancel. The cost side is equally problematic, as multi-call systems multiply per-question latency and API expenditure. FaiTH's response is architectural. The pipeline has exactly two LLM call sites, one mandatory compile call in Stage 1 and one optional reason call in Stage 4, and the deterministic shortcut (Eq. (14)) eliminates the second call whenever the symbolic engine has already isolated an unambiguous answer. All intermediate steps, including grounding, retrieval, filtering, alignment, composition, and recovery, are deterministic and introduce no sampling noise.

Design motivations for heterogeneity. A further motivation is the schema heterogeneity of existing benchmarks, which differ in time representation (point timestamps vs. validity intervals), granularity (day, month, year), and the presence of sentinel endpoints for open-ended facts. Prior systems typically hard-code one representation. FaiTH absorbs this heterogeneity in two places only, namely the extended time domain $\mathcal{T}^\star$ (Eq. (1)) with the unified quintuple view of facts, and the benchmark-specific rule lists of the grounding cascade. Every operator downstream of these two interfaces is shared verbatim across datasets.

B.2 Notation Table

Table 5 summarizes all symbols introduced in Section 4, grouped by stage in which they first appear.

Table 5: Notation used throughout the paper, grouped by the stage in which each symbol first appears.

Group	Symbol	Meaning
Preliminaries	$\mathcal{G}$	Temporal knowledge graph $(\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F})$ of entities, relations, timestamps, and facts.
	$(s, r, o, \tau_s, \tau_e)$	Unified quintuple view of a fact with validity interval $[\tau_s, \tau_e]$.
	$\mathcal{T}^\star$	Extended time domain $\mathcal{T} \cup \{\tau_{-\infty}, \tau_{+\infty}\}$ (Eq. (1)).
	$\preceq$	Total order on $\mathcal{T}^\star$ used by all temporal comparisons.
	$\tau_{-\infty}, \tau_{+\infty}$	Sentinel endpoints for open-ended validity intervals.
	$\lfloor\cdot\rfloor_\gamma$	Truncation of a timestamp to granularity γ.
Stage 1	$\mathcal{C}_q$	Compiled constraint specification of question q (Eq. (2)).
	$\Psi = \{\psi_i\}_{i=1}^k$	Event templates, $k \in \{1, 2, 3\}$, each $\psi_i = (s_i, r_i, o_i)$.
	$\alpha \in \mathcal{A}$	Answer slot drawn from the nine-element inventory of Eq. (3).
	$\Omega = [\tau_s^q, \tau_e^q]$	Single-event temporal window, possibly $\perp$.
	$\Lambda = \{\lambda_{ij}\}$	Inter-event relations over template pairs, active when $k \geq 2$.
	$\mathcal{L}$	Six predicates projected from the thirteen Allen relations (Eq. (4)).
	θ	Temporal operator selecting timeline arithmetic.
	γ	Evaluation granularity, $\gamma \in \{\mathsf{day}, \mathsf{month}, \mathsf{year}\}$.
Stage 2	ϕ_E, ϕ_R	Deterministic entity and relation grounders.
	Φ	Specification-level grounding operator (Eq. (5)).
	$\hat{\mathcal{C}}_q, \hat{\Psi}, \hat{\psi}_i$	Grounded specification, grounded template set, and template.
	μ_ℓ, w_ℓ	Cascade matching rules and their confidence weights (Eq. (6)).
	$\chi(\hat{e}, \hat{r})$	Constant-time co-occurrence probe over inverted indices (Eq. (7)).
	$\perp$	Failure value of entity grounding.
Stage 3	$\rho, \bar{\rho}, \pi_{\bar{\rho}}(f)$	Unknown role, complementary role, and role projection of a fact.
	Query	Role-aware retrieval operator (Eq. (8)).
	Filter_γ	Granularity-aware single-event temporal filter (Eq. (9)).
	Align_λ	Inter-event alignment operator with envelope endpoints (Eq. (10)).
	$\mathrm{Compose}_\theta, \Theta_\theta$	Multi-event composition and its arithmetic function (Eq. (11)).
	$\mathcal{S}, \mathcal{S}_a, \mathcal{S}_r$	Structurally consistent set, anchor set, and reference set; $\mathcal{S}_{\mathrm{ref}}$ unions the non-anchor retrievals.
	$\mathcal{F}_q, \beta$	Faithful evidence set (Eq. (12)) and its adaptive size cap.
	$\mathrm{Cand}, \mathrm{Boundary}_\gamma$	Candidate list and boundary extractor, dispatched over the answer-slot partitions $\mathcal{A}_{\mathrm{bnd}}, \mathcal{A}_{\mathrm{cmp}}$ of Eq. (13).
Stage 4	B, B'	Per-question LLM budget before and after Stage 4 ($B \leq 2$).
	$\mathsf{SortDirected}(\hat{\mathcal{C}}_q)$	Predicate marking an ordering directive such as "first" or "last".
	$\mathsf{Shortcut}$	Deterministic shortcut condition (Eq. (14)).
	$\textsc{Reason}, \hat{a}$	Constrained reasoning call and final answer (Eq. (15)).
Appendices	$\mathcal{R}_{\mathrm{rec}}$	Ordered rewrite sequence of the symbolic recovery (Appendix B.5).
	$c_{\mathrm{LLM}}, d_{\max}$	Cost of one LLM call and maximum entity degree (Appendix B.8).

B.3 Instantiation Across Benchmarks

FaiTH runs the same four-stage pipeline on all three benchmarks. This subsection records benchmark-specific instantiations, all of which live in the Stage-1 prompt, the Stage-2 rule lists, and the answer-format instructions of the Stage-4 reason prompt, leaving Stage 3 unchanged across benchmarks.

Stage 1. The compile prompt keeps the same four-stage contract on every benchmark but differs in its JSON fields, its few-shot examples, and the enumeration of native question types. For MULTITQ, the prompt lists the dataset's question types (*equal*, *before_after*, *first_last*, *equal_multi*, *after_first*, *before_last*) and instructs the compiler to emit granularity $\gamma \in \{\mathsf{day}, \mathsf{month}, \mathsf{year}\}$ matching the question surface. For the two TIMELINEKGQA benchmarks, the prompt instead enumerates the interval-oriented answer slots (tsr, dur, rdur, rank, setop) and explains the sentinel conventions *beginning of time* and *end of time*, which the compiler maps to $\tau_{-\infty}$ and $\tau_{+\infty}$. The two post-processing steps use the same trigger lexicon on all benchmarks.

Stage 2. The cascade skeleton of Eq. (6) is fixed, and only the rule lists $\{\mu_\ell\}$ adapt to each TKG's naming convention. On MULTITQ, whose entities follow the ICEWS style with parenthesized country

qualifiers, the entity cascade ϕ_E proceeds through exact match, case-insensitive match, underscore and space normalization, and a schema-aware *Role (Country)* pattern match that recognizes identifiers such as *Head_of_Government_(Afghanistan)*. A reference-resolution refinement ϕ_E^{ref} re-anchors pronoun-like mentions (*this country*, *the same leader*) to the nearest previously grounded entity of a compatible type. On the TIMELINEKGQA benchmarks, whose entities are natural-language phrases, the cascade keeps the string-normalization and similarity tiers and drops the ICEWS-specific schema tier. The relation cascade ϕ_R maps surface predicates to canonical relation identifiers with per-benchmark synonym tables and returns an ordered candidate list, which the probe χ of Eq. (7) prunes against pre-built inverted indices. The indices are constructed once per benchmark in a single offline pass over $\mathcal{F}$ and store, for every entity, the set of relations it participates in for each role.

Stage 3. All four operators are shared. The only benchmark-dependent behavior is induced by data characteristics. On MULTITQ, facts are point-valued, so every interval degenerates to $[\tau, \tau]$ and the envelope endpoints of Eq. (10) coincide with single timestamps. On TIMELINEKGQA, intervals are genuine and may carry sentinel endpoints, which the total order $\preceq$ on $\mathcal{T}^\star$ handles without case distinctions. Granularity truncation $\lfloor \cdot \rfloor_\gamma$ is exercised chiefly by MULTITQ, whose questions mix day, month, and year granularities.

Stage 4. The shortcut predicate is shared, and the reason prompt keeps the same structure. The reason prompt renders $\mathcal{F}_q$ chronologically, one fact per line, using each benchmark's native timestamp format, and instructs the model to answer strictly from the rendered facts. On TIMELINEKGQA, the prompt additionally explains how to phrase interval answers and how to treat sentinel endpoints, as shown in Appendix D.

B.4 ALLEN RELATION PROJECTION

Table 6 enumerates the projection from the thirteen Allen interval relations (Allen, 1983) onto the six temporal predicates of $\mathcal{L}$ (Eq. (4)) used by the inter-event alignment operator. The projection collapses distinctions that the questions of these benchmarks rarely lexicalize, for example *meets* versus *before* and *contains* versus *during*, and keeps the ordering and coincidence distinctions that the temporal connectives in the questions do express.

Table 6: Projection of the thirteen Allen interval relations onto the six FaiTH temporal predicates in $\mathcal{L}$, together with typical surface signals that trigger each predicate during Stage-1 compilation.

Allen relation(s)	FaiTH predicate	Typical surface signal
before / after	before / after	"before", "after", "prior to", "following"
meets / met-by	before / after	"right before", "immediately after"
overlaps / overlapped-by	during	"while", "overlapping with"
during / contains	during	"during", "within", "throughout"
starts / started-by	startsame	"started at the same time as"
finishes / finished-by	finishsame	"ended at the same time as"
equal	sametime	"at the same time", "simultaneously"

B.5 RECOVERY SUBROUTINE DETAILS

When Stage 3 returns an empty evidence set, FaiTH applies a symbolic recovery subroutine $\mathcal{R}_{\text{rec}}$ consisting of an ordered sequence of deterministic rewrites to $\hat{\mathcal{C}}_q$, re-executing Stage 3 after each rewrite and stopping at the first non-empty result. The rewrites are ordered from least to most invasive, and none of them consumes LLM budget.

Re-grounding with alternative candidates. The first family of rewrites revisits Stage-2 decisions. For each grounded slot, the cascade's next-best candidates (up to four per slot) are substituted one at a time, in descending confidence order. This repairs cases where the top-ranked grounding was a near-miss, for example a demonym resolved to the wrong country office, while keeping every other field of the specification intact.

Role and constraint relaxation. The second family relaxes structural and temporal constraints that the compiler may have over-specified. The role restriction of Eq. (8) is flipped to the complementary role, which repairs residual subject and object confusions that survived the passive-voice role swap. If the specification carries a temporal window Ω whose granularity is finer than the benchmark's native resolution, the window is widened by one granularity level. If an inter-event relation in Λ yields an empty intersection, it is dropped and the anchor template is re-executed as a single-event query.

Relation broadening. The third family widens the relation scope. The probe-filtered candidate list of Stage 2 is replaced by the full pre-probe list from ϕ_R, and if the list still yields no facts, the retrieval falls back to all relations incident to the anchor entity, with the temporal constraints of the specification still enforced. This final rewrite trades structural precision for recall, and the facts it returns still pass through the same temporal filtering and alignment, so temporal consistency in the sense of Definition 2 is preserved even in the last resort.

B.6 Faithfulness Proofs

This appendix proves the two theorems of Section 4.3. Both are conditional on Assumption 1.

Proof of Theorem 1. Part (i), structural consistency of retrieval. Let $f = (s, r, o, \tau_s, \tau_e) \in \mathrm{Query}(\hat{\psi}_i, \rho)$. By the set definition in Eq. (8), f satisfies both membership conditions $r = \hat{r}_i$ and $\pi_{\bar{\rho}}(f) = \hat{e}_i$. The first condition is exactly the relation clause of Definition 1. For the slot clause, consider the two cases of the unknown role ρ. If $\rho = \mathsf{subj}$, then the template has $\hat{s}_i = ?$ and a non-placeholder object $\hat{o}_i = \hat{e}_i$, the complementary role is $\bar{\rho} = \mathsf{obj}$, and $\pi_{\bar{\rho}}(f) = o$, so the membership condition gives $o = \hat{e}_i = \hat{o}_i$, which is the only non-placeholder slot to check. If $\rho = \mathsf{obj}$, the symmetric argument gives $s = \hat{e}_i = \hat{s}_i$. In both cases every non-placeholder slot of $\hat{\psi}_i$ coincides with the corresponding element of f, so f is structurally consistent with $\hat{\psi}_i$. Under Assumption 1, $\hat{e}_i$ and $\hat{r}_i$ are the identifiers intended by the question, hence consistency holds with respect to the intended template rather than a mis-grounded surrogate.

Part (ii), temporal consistency of filtering and alignment. Consider first the single-event window. If $\Omega = [\tau_s^q, \tau_e^q] \neq \bot$, membership in $\mathrm{Filter}_\gamma(\mathcal{S}, \Omega)$ of Eq. (9) requires

$$\lfloor \tau_s^q \rfloor_\gamma \preceq \lfloor \tau_s^{(f)} \rfloor_\gamma \quad \text{and} \quad \lfloor \tau_e^{(f)} \rfloor_\gamma \preceq \lfloor \tau_e^q \rfloor_\gamma,$$

which are identical to the window conditions of Definition 2 at granularity γ. Since $\preceq$ is a total order on $\mathcal{T}^\star$ and truncation $\lfloor \cdot \rfloor_\gamma$ is monotone with respect to $\preceq$, the comparison is well defined for every fact, including facts carrying sentinel endpoints, and the degenerate windows $[\tau, \tau]$, $[\tau_{-\infty}, \tau)$, and $(\tau, \tau_{+\infty}]$ realize the equal, before, and after constraints as special cases.

Consider next the inter-event relations for $k \geq 2$. Let $f \in \mathrm{Align}_\lambda(\mathcal{S}_a, \mathcal{S}_r)$ with envelope endpoints $\tau_s^r = \min_{g \in \mathcal{S}_r} \tau_s^{(g)}$ and $\tau_e^r = \max_{g \in \mathcal{S}_r} \tau_e^{(g)}$. Membership in Eq. (10) requires $\mathrm{rel}_\lambda(\tau_s^{(f)}, \tau_e^{(f)}, \tau_s^r, \tau_e^r)$ to hold, where the six predicates are

- before: $\tau_e^{(f)} \preceq \tau_s^r$;
- after: $\tau_e^r \preceq \tau_s^{(f)}$;
- during: $\tau_s^r \preceq \tau_s^{(f)}$ and $\tau_e^{(f)} \preceq \tau_e^r$;
- sametime: $\lfloor \tau_s^{(f)} \rfloor_\gamma = \lfloor \tau_s^r \rfloor_\gamma$ and $\lfloor \tau_e^{(f)} \rfloor_\gamma = \lfloor \tau_e^r \rfloor_\gamma$;
- finishsame: $\lfloor \tau_e^{(f)} \rfloor_\gamma = \lfloor \tau_e^r \rfloor_\gamma$;
- startsame: $\lfloor \tau_s^{(f)} \rfloor_\gamma = \lfloor \tau_s^r \rfloor_\gamma$.

Each predicate is precisely the truth condition that Definition 2 assigns to the corresponding $\lambda \in \mathcal{L}$, evaluated between the anchor fact and the envelope of the reference facts, with every comparison under $\preceq$ on $\mathcal{T}^\star$. Facts violating the assigned relation fail the predicate and are excluded, therefore every surviving fact satisfies all active clauses of Definition 2. □

Proof of Theorem 2. Definition 3 has a soundness and a completeness clause; we verify each.

Soundness. Every $f \in \mathcal{F}_q$ has passed through the operator composition

$$\mathrm{Align}_\Lambda \circ \mathrm{Filter}_\gamma \circ \bigcup_{i=1}^{k} \mathrm{Query}(\hat{\psi}_i, \rho_i)$$

followed by the cap $|_{\leq\beta}$, which only removes elements and never adds any. Membership in the union of retrievals implies, by Theorem 1(i), that f is structurally consistent with some $\hat{\psi}_i \in \hat{\Psi}$. Survival through Filter_γ and Align_Λ implies, by Theorem 1(ii), that f is temporally consistent with $\hat{\mathcal{C}}_q$. Therefore, every member in the evidence set $\mathcal{F}_q$ satisfies both consistency requirements.

Completeness. Let $f \in \mathcal{F}$ be any fact that is structurally consistent with some $\hat{\psi}_i$ and temporally consistent with $\hat{\mathcal{C}}_q$. Structural consistency gives $r = \hat{r}_i$ together with agreement on every non-placeholder slot, which is exactly the membership condition of $\mathrm{Query}(\hat{\psi}_i, \rho_i)$, so f enters the union of retrievals. Temporal consistency means f satisfies the window condition and every active inter-event predicate, so f survives Filter_γ and Align_Λ. Therefore f belongs to the uncapped composition. The final cap $|_{\leq\beta}$ either retains f, in which case $f \in \mathcal{F}_q$, or excludes it, which by the definition of the adaptive bound happens only when f is tied at a temporal boundary position whose tie group has already saturated the bound. This is precisely the exception permitted by the completeness clause of Definition 3. Therefore, $\mathcal{F}_q$ is faithful. □

B.7 Complete Pipeline Pseudo-code

Algorithm 1 presents FaiTH's pseudo-code, including the deterministic post-processing of Stage 1, the cascaded grounding with probing and inverse-relation retry of Stage 2, the operator composition with adaptive capping and budget-neutral recovery of Stage 3, and the shortcut dispatch of Stage 4. The two LLM calls leveraged to answer each question are marked explicitly on the right margin.

Algorithm 1 Pseudo-code of FaiTH.

Require: TKG $\mathcal{G}$ with inverted indices; question q; budget $B \leftarrow 2$; cap β; rewrites $\mathcal{R}_{\mathrm{rec}}$
Ensure: answer $\hat{a}$

{**Stage 1: Compile** (Eqs. (1) to (4))}
$J \leftarrow$ Compile-LLM(q); parse J into specification $\mathcal{C}_q = (\Psi, \alpha, \Omega, \Lambda, \theta, \gamma)$; $B \leftarrow B - 1$ ▷ LLM call 1
repair: recover missing Λ or ordering from surface cues ("before", "after", "first", "last"); swap subject and object for passive-voice predicates
{**Stage 2: Ground** (Eqs. (5) to (7)), no LLM budget consumed}
for each event template $\psi_i = (s_i, r_i, o_i) \in \Psi$ **do**
    ground each non-placeholder entity slot via ϕ_E, taking the candidate of the first firing rule μ_ℓ
    ground r_i via ϕ_R into candidates $[\hat{r}_{i,1}, \ldots, \hat{r}_{i,m}]$; retain only those passing probe $\chi(\hat{e}_i, \hat{r}_{i,j})$
    if all forward probes fail, retry with inverse relations; if both fail, mark the template ungrounded
end for
assemble grounded specification $\hat{\mathcal{C}}_q = (\hat{\Psi}, \alpha, \Omega, \Lambda, \theta, \gamma)$
{**Stage 3: Execute** (Eqs. (8) to (13)), no LLM budget consumed}
$\mathcal{S} \leftarrow \bigcup_i \mathrm{Query}(\hat{\psi}_i, \rho_i)$ with role restriction $\pi_{\bar{\rho}}(f) = \hat{e}_i$; retry the complementary role if empty
if $\Omega \neq \bot$ **then** $\mathcal{S} \leftarrow \mathrm{Filter}_\gamma(\mathcal{S}, \Omega)$, comparing under $\preceq$ on $\mathcal{T}^\star$ after truncation $\lfloor\cdot\rfloor_\gamma$
if $k \geq 2$ **then for** each $\lambda_{ij} \in \Lambda$: compute reference envelope (τ_s^r, τ_e^r); $\mathcal{S} \leftarrow \mathrm{Align}_{\lambda_{ij}}(\mathcal{S}, \mathcal{S}_{\mathrm{ref}})$
$\mathcal{F}_q \leftarrow \mathcal{S}\,|_{\leq\beta}$, ranked by the temporal key from (α, θ) and preserving all facts tied at the boundary
if $\mathcal{F}_q = \emptyset$ **then**
    for each rewrite $\xi \in \mathcal{R}_{\mathrm{rec}}$ (re-grounding, relaxation, broadening): re-execute $\xi(\hat{\mathcal{C}}_q)$, stop if non-empty
end if
if $\alpha \in \mathcal{A}_{\mathrm{bnd}}$ **then** $\mathrm{Cand} \leftarrow \mathrm{Boundary}_\gamma(\mathcal{F}_q, \alpha, \theta)$ **else** $\mathrm{Cand} \leftarrow \{\mathrm{Compose}_\theta(f_1, \ldots, f_k)\}$ by Eq. (13)
verification: re-check existence and temporal membership of every fact in $\mathcal{F}_q$
{**Stage 4: Reason** (Eqs. (14) to (15))}
if $|\mathrm{Cand}| = 1$ **or** SortDirected$(\hat{\mathcal{C}}_q)$ **then**
    return $\hat{a} \leftarrow \mathrm{top}(\mathrm{Cand})$ ▷ deterministic shortcut, no further LLM call
else
    render $\mathcal{F}_q$ chronologically; $\hat{a} \leftarrow$ Reason-LLM$(q, \mathcal{F}_q, \hat{\mathcal{C}}_q)$; $B \leftarrow B - 1$ ▷ LLM call 2
    return $\hat{a}$
end if

B.8 Time Complexity Analysis

Let c_{LLM} denote the cost of one LLM call, $d_{\max}$ the maximum number of facts incident to any entity, and $k \leq 3$ the number of event templates. Stage 1 costs one LLM call plus linear-time JSON parsing and repair, contributing $O(c_{\mathrm{LLM}})$. Stage 2 evaluates the cascade on a constant number of mentions. Each rule is a dictionary or index lookup over pre-built structures, and the probe of Eq. (7) is a constant-time set membership on the inverted index, so the stage costs $O(1)$ per mention after the one-off $O(|\mathcal{F}|)$ index construction shared across all questions. Stage 3 retrieves at most $d_{\max}$ facts per template through the inverted index, filters and aligns them in linear time, and sorts them by the temporal key for boundary extraction and capping, giving $O(k \cdot d_{\max} \log d_{\max})$. The recovery subroutine re-executes this bound a constant number of times in the worst case. Stage 4 either returns in constant time through the shortcut or issues the single reason call, contributing at most $O(c_{\mathrm{LLM}})$. The end-to-end complexity is therefore

$$O(c_{\mathrm{LLM}} + k \cdot d_{\max} \log d_{\max}) \,,$$

dominated in practice by the one to two LLM calls, since the symbolic operators execute in milliseconds on commodity hardware.

C Extended Experimental Details

C.1 Benchmark and Baseline Descriptions

C.1.1 Benchmarks

The MultiTQ (Chen et al., 2023) benchmark, released but without explicitly specifying the license, is the largest publicly available TKGQA benchmark with point-valued timestamps. It is constructed from ICEWS05-15 (Boschee et al., 2015), a TKG recording political, economic, and social events at day-level granularity between 2005 and 2015. The underlying temporal knowledge graph contains roughly 10,500 entities, 251 relations, and 480,000 quadruples, with a semantic complexity substantially higher than that of Wikidata-based TKGQA benchmarks (Chen et al., 2023). MultiTQ comprises approximately 500,000 question-answer pairs spanning three temporal granularities (*day*, *month*, *year*) and six question categories, namely three single-constraint types (*equal*, *before_after*, *first_last*) and three multiple-constraint types (*equal_multi*, *after_first*, *before_last*). An important principle of MultiTQ is the elimination of pseudo-temporal questions, indicating that every question genuinely requires temporal reasoning to be answered correctly. The TimelineKGQA-Cron benchmark, released under the MIT license, is constructed from the CronQuestions KG (Saxena et al., 2021) using the TimelineKGQA generator (Sun et al., 2025). Unlike MultiTQ, facts in this benchmark are interval-valued quintuples, while questions are stratified into three difficulty levels (*simple*, *medium*, *complex*) according to the number of events involved, and into nine answer slot types that include entity answers (*subject*, *object*), boundary times (*timestamp_start*, *timestamp_end*), intervals (*timestamp_range*), durations (*duration*, *relation_duration*), ranked positions (*relation_ranking*), and set-valued answers (*relation_union_or_intersection*). This diversity forces a TKGQA system to go beyond retrieval over question-related facts and to support genuine temporal arithmetic such as duration summation, interval intersection, and Allen-style relation comparison between event pairs (Allen, 1983). TimelineKGQA-ICEWS, released under the MIT license, is also generated by the TimelineKGQA pipeline (Sun et al., 2025) but over the ICEWS Actor dictionary (Boschee et al., 2015). All facts in this benchmark share a single canonical relation `Affiliation To`, therefore the challenge shifts from relation disambiguation to precise entity linking and fine-grained interval reasoning. Entities follow the naming convention of ICEWS and are augmented with curated aliases, while intervals may contain sentinel endpoints such as *beginning of time* and *end of time* that represent open-ended affiliations. The answer slot inventory matches that of TimelineKGQA-Cron, and questions span the same difficulty stratification (*simple*, *medium*, *complex*), but they additionally exercise the full thirteen Allen temporal relations (Allen, 1983) between event intervals, including subtle cases such as *finishes*, *started-by*, and *contains*.

C.1.2 Baselines

Pre-trained language models. BERT (Devlin et al., 2019) generates question embeddings that are concatenated with entity and temporal embeddings and passed through a learnable projection

layer (Chen et al., 2023). We also include LLaMA2-7B (Touvron et al., 2023) and ChatGPT (OpenAI, 2023), both evaluated by directly feeding questions without additional context.

TKG embedding and GNN methods. EmbedKGQA (Saxena et al., 2020) applies static embeddings to temporal settings without explicit temporal modeling. CronKGQA (Saxena et al., 2021) combines temporal KG embeddings with BERT-based question representations. MultiQA (Chen et al., 2023) extends CronKGQA with a multi-granularity time aggregation module that fuses day-level embeddings into coarser representations using a transformer encoder (Vaswani et al., 2023). These methods score candidate answers through embedding similarity and remain limited to the answer vocabularies defined by the TKG.

LLM-based methods with prompt engineering. ARI (Chen et al., 2024b) augments question understanding through adaptive reasoning instructions. RTQA (Gong et al., 2025) recursively decomposes complex temporal questions into a tree of sub-questions solved bottom-up, and employs multi-path answer aggregation that combines answers from both sub-questions and the original question to mitigate error propagation. RTQA uses GPT-4o-mini (OpenAI, 2024b) for decomposition and DeepSeek-V3 (Liu et al., 2024) for reasoning, averaging 3.96 API calls per question on MULTITQ. On the two TimelineKGQA benchmarks, we additionally include a RAG baseline (Lewis et al., 2020) that encodes both the TKG and queries using a dense embedding model and retrieves the top-1 most relevant fact, as well as GPT-4o (OpenAI, 2024a) evaluated by directly feeding questions.

LLM fine-tuning. TimeR^4 (Qian et al., 2024) performs question rewriting to expose implicit temporal constraints using GPT-3.5 Turbo (OpenAI, 2023), paired with a fine-tuned Sentence-BERT (Reimers & Gurevych, 2019) retriever and a fine-tuned Llama-2-7B (Touvron et al., 2023) generator. PoK (Qian et al., 2025) extends TimeR^4 with a plan-of-knowledge framework that decomposes questions into structured sub-objectives (Retrieve, Rank, Reason) by prompting GPT-4o (OpenAI, 2024a), and trains a contrastive time-aware retriever with an InfoNCE objective (Rusak et al., 2025) on hard negatives made by corrupting entities, relations, or timestamps.

Reinforcement learning. Temp-R1 (Gong et al., 2026) formulates TKGQA as a Markov decision process and fine-tunes Llama-3.1-8B (Grattafiori et al., 2024) with group relative policy optimization (GRPO) (Guo et al., 2025). It expands the action space with internal actions (`<plan>`, `<filter>`, `<rank>`) alongside an external `<search>` action, and introduces reverse curriculum learning that trains on difficult questions first to prevent shortcut behavior. Its supervised fine-tuning cold-start data is generated by GPT-4o at a one-time cost of approximately $50, but inference is fully local. Search-R1 (Jin et al., 2025) applies similar reinforcement learning principles to teach models when and how to invoke external search tools.

C.2 Evaluation and Implementation Details

Following prior work (Gong et al., 2025), we adopt Hits@k as the evaluation metric. Hits@k measures the proportion of questions for which a correct answer appears within the top-k predictions, where correctness is determined by type-aware string matching, namely entity answers are compared after text normalization by lenient matching (string containment, and on TL-ICEWS additionally a character-similarity threshold), while temporal answers are parsed into their native forms before comparison. We primarily report Hits@1, which directly reflects a framework's ability to produce the correct answer as the top prediction, and selectively report Hits@10. For FaiTH, the top-k list is read off the ordered candidate list of Stage 3 when the deterministic shortcut fires, and from the constrained LLM output otherwise, with timestamp answers compared at the granularity γ of the specification. The temperature for making LLM calls is set to 0. For the sensitivity analysis of Section 5.5, the nine answer slots of TL-ICEWS are grouped into four size-weighted buckets: *Entity* (subj, obj), *Time* (tss, tse, tsr), *Duration* (dur, rdur), and *Relation* (rank, setop). Even at temperature 0, a cloud LLM retains some non-determinism due to non-associative floating-point accumulation in batched inference or provider-side infrastructure changes such as engine upgrades and quantization adjustments. Four independent full test set runs of FaiTH (DS-V4-Flash) on MULTITQ score 81.93%, 81.70%, 81.64%, and 81.42% overall Hits@1, a spread of 0.51 points. Tables 1, 3, and 10 all report the 81.70% run, so every headline number is a single run rather than the best of several or an average.

Each analysis below is likewise pinned to one run, identified by the baseline column of its table, so a baseline and its perturbations share a run.

C.3 LLM CALL BUDGET AND API COST

Table 7 reports the per-question LLM call distribution and the estimated API cost of FaiTH under cloud-based LLMs on all three benchmarks.

Table 7: Per-question LLM call count and estimated API cost of FaiTH on the three benchmark test sets, under two generative LLM backbones: DeepSeek-V3.2 (DS-V3.2) and GPT-5.4. Each question consumes at most two LLM calls by construction: one for the *compile* step of Stage 1 and, when the deterministic shortcut of Stage 4 is not triggered, one for the *reason* step of Stage 4.

			1 LLM Call		2 LLM Calls		
TKGQA Dataset	Backbone	Total	Count	Ratio	Count	Ratio	API Cost
MULTITQ	DS-V3.2	54,584	35,761	65.52%	18,823	34.48%	$\sim$ \$20
	GPT-5.4	54,584	35,292	64.66%	19,292	35.34%	$\sim$ \$145
TIMELINEKGQA-CRON	DS-V3.2	8,344	5,779	69.26%	2,565	30.74%	$\sim$ \$3
	GPT-5.4	8,344	5,781	69.28%	2,563	30.72%	$\sim$ \$24
TIMELINEKGQA-ICEWS	DS-V3.2	14,621	10,472	71.62%	4,149	28.38%	$\sim$ \$5
	GPT-5.4	14,621	10,357	70.84%	4,264	29.16%	$\sim$ \$39

C.4 CASE STUDIES: HOW FAITHFULNESS DELIVERS TKGQA CORRECTNESS

We present a qualitative study that makes the faithfulness commitments concrete on four representative MULTITQ questions. Since every specification on MULTITQ collapses to $k = 1$, the symbolic engine of Stage 3 only exercises Query and Filter_γ. Every observation below can be attributed to these two operators and their interaction with the Stage 4 deterministic shortcut. We contrast the full pipeline against the two ablations most directly tied to Stage 3, namely *w/o role-aware* (which removes the role restriction $\pi_{\bar{\rho}}(f) = \hat{e}_i$ in Query) and *w/o temporal filter* (which removes the granularity-aware window in Filter_γ). Table 8 summarizes the execution traces of the four cases.

Table 8: Execution traces of four MULTITQ cases under the full FaiTH (DS-V3.2) pipeline and the two Stage-3 ablations. For each case, we report the evidence set size $|\mathcal{F}_q|$ returned by Stage 3 before the adaptive bound of Eq. (12) is applied, the Hits@1 judgment (✓ correct, × incorrect), and concise failure modes introduced by the ablation. FaiTH reaches the deterministic shortcut of Eq. (14) on all four cases with a single Stage-1 LLM call.

		FaiTH, full		w/o role-aware			w/o temporal filter		
Case	Type	$\lvert\mathcal{F}_q\rvert$	H@1	$\lvert\mathcal{F}_q\rvert$	H@1	Failure mode	$\lvert\mathcal{F}_q\rvert$	H@1	Failure mode
C1	entity	1	✓	6	×	role-inverted facts	20	×	window lost, mis-ranked
C2	entity	1	✓	2	✓	*happened to be correct*	40	×	time slice unresolved
C3	time	1	×	1	×	spurious Ω enforced	2	✓	*happened to be correct*
C4	entity	1	✓	2	×	role-inverted facts	40	×	time slice unresolved

C1: Query **secures structural consistency.** The first case asks *"After the Kurdistan Workers' Party guerrillas, against which country did Iraq first use conventional military force?"*, whose gold answer is France. The question places Iraq at the *subject* role of the target fact, and any retrieval that returns Iraq as the *object* of a military action would supply a fact in which Iraq is the target rather than the actor. The full pipeline compiles $\rho = \mathsf{obj}$, with Iraq as the subject anchor, and executes Query with the role restriction of Eq. (8), producing a single structurally consistent fact and triggering the Stage 4 shortcut. Removing the role restriction retrieves all facts in which Iraq appears in either role, which adds five role-inverted facts to the evidence set and causes the reasoner to select a country in which Iraq is the *target* of force. This is exactly the failure mode that Definition 1 rules out by construction.

C2: Filter_γ **for temporal consistency.** The second case asks *"On 16 October 2010, which country sought diplomatic cooperation with China?"*, whose gold answer is Vietnam. Diplomatic cooperation with China is a dense, recurring event over the ICEWS05-15 period, so Query alone retrieves dozens of role-consistent facts spanning across many years. The full pipeline applies Filter_γ with $\Omega = [2010\text{-}10\text{-}16, 2010\text{-}10\text{-}16]$ on the daily granularity and collapses the candidate set to a single fact, triggering the deterministic shortcut in Stage 4. Removing the temporal filter leaves forty role-consistent facts in the evidence set, and the reasoner, unable to isolate the correct time slice, fails to produce the correct country name as its top prediction. This is a direct instantiation of Challenge 1 in Appendix B.1: temporal unfaithfulness cannot be recovered by the reasoner alone, because the prompt no longer contains a structurally isolable signal for the queried time.

C3: Understanding the *w/o tf* gain on Time-type questions. Table 4 shows that removing the temporal filter on MULTITQ *increases* the Time-type answer score from 0.847 to 0.911, which seems to be counter-intuitive. The third case explains this phenomenon transparently. The question *"In which year did the presidential family of the United States last visit China?"* is a first_last question whose correct treatment requires only a descending sort over the anchor facts and no absolute temporal window Ω should be involved. The compilation step, however, occasionally mis-classifies such questions as equal_multi and synthesizes a spurious reference time field. In the full pipeline, this spurious field is promoted to Ω and Filter_γ faithfully enforces it, pruning the correct fact in 2009 in favor of the misleading fact in 2005. Removing Filter_γ neutralizes the error brought by Stage 1 and allows the descending sort to reach the gold answer. We discuss this case to hopefully refine the interpretation of the *w/o tf* column of Table 4: the Time-type answer score increase is not valid evidence that temporal filtering could be harmful for FaiTH on MULTITQ, but rather that Filter_γ amplifies Stage-1 compilation noise on a small subset of first_last questions in which Ω is spuriously present. On the entity side, the same ablation causes a significant drop ($0.817 \rightarrow 0.342$) because the noise channel is dwarfed by the loss of temporal isolation illustrated in C2.

C4: Query **and** Filter_γ **are independently necessary.** The fourth case asks *"Who made a request to Thailand in 2008-12?"*, whose gold answer is the ICEWS-style entity name as `Citizen (Thailand)`. The full pipeline compiles $\rho = \mathsf{subj}$, $\Omega = [2008\text{-}12, 2008\text{-}12]$ on the monthly granularity, and recovers a single fact through the composition of Query and Filter_γ. The two ablations fail along orthogonal axes. Removing the role restriction admits facts in which Thailand is the subject of requests to other parties, and the resulting two-candidate pool causes the reasoner to select the incorrect entity candidate. Removing the temporal filter retains role-consistent facts but loses the time constraint, so the reasoner receives a forty-fact evidence pool and again fails to select the correct entity. The joint observation aligns well with Definition 3, where a faithful evidence set must require structural and temporal consistency simultaneously. Neither operator is, on its own, sufficient.

We discuss the four case studies to argue that FaiTH's empirical gains are a direct consequence of the faithfulness definitions introduced and achieved in Section 4.3. Query (Eq. (8)) enforces structural consistency and prevents role-inverted facts from entering the evidence set (C1, C4). Filter_γ (Eq. (9)) enforces temporal consistency and collapses the candidate set so that the Stage 4 shortcut fires (C2, C4). The *w/o tf* anomaly on Time-type questions is an amplification of Stage-1 compilation noise rather than evidence against temporal filtering itself (C3). Taken together, these cases illustrate that structural and temporal faithfulness are not post-hoc desiderata but the mechanism that turns a correctly compiled question into the correct answer, as observed in C1, C2, and C4.

C.5 LLM BACKBONE SENSITIVITY

This appendix reports the per-question-type breakdown behind Table 3 in Section 5.3. We evaluate FaiTH under 11 LLM backbone configurations spanning 9 base models from 5 model families: three cloud-hosted LLMs named in Section 5.1 and eight open-weight deployments served by Ollama on one Apple Mac Studio. Table 9 lists the local models. Every configuration uses identical prompts, symbolic engine, and evaluation scripts. We include the strongest baseline, Temp-R1 (Gong et al., 2026), for reference. Tables 10 to 12 show that quantized local models can match or exceed cloud-hosted LLMs: Gemma4:31b-nvfp4 (20 GB) achieves the best overall score on TL-ICEWS with 69.71%, ahead of both DS-V3.2 and GPT-5.4, and GLM-4.7-Flash:q4_K_M (19 GB) reaches 70.23% on TL-CRON, close behind DS-V3.2.

Table 9: Dense and mixture-of-experts (MoE) LLM backbones under various types of quantization. All locally deployable LLMs for FaiTH are implemented on one Apple Mac Studio with M4 Max having 36 GB unified memory through Ollama with temperature 0.

Model Tag	Family	Architecture	Size
Gemma4:31b-nvfp4	Gemma 4	Dense	20 GB
Gemma4:26b-mxfp8	Gemma 4	MoE	26 GB
Gemma4:12b-mlx-bf16	Gemma 4	Dense	23 GB
Qwen3.6:27b-nvfp4	Qwen 3.6	Dense	19 GB
Qwen3.6:27b-mtp-q8_0	Qwen 3.6	Dense	29 GB
Qwen3.6:35b-a3b-nvfp4	Qwen 3.6	MoE	21 GB
GLM-4.7-Flash:q4_K_M	GLM-4.7-Flash	MoE	19 GB
GLM-4.7-Flash:q8_0	GLM-4.7-Flash	MoE	31 GB

Table 10: FaiTH on MULTITQ (full test set, Hits@1 %). Best and second-best FaiTH configuration per column are in **bold** and *italic*; Temp-R1 is listed for reference and is not ranked.

#	Backbone	Deploy	Overall	Single	Multiple	Entity	Time
1	FaiTH (GPT-5.4)	Cloud	**85.20**	**90.50**	*72.20*	**83.00**	**90.80**
2	FaiTH (DS-V3.2)	Cloud	*82.60*	*87.00*	71.60	*81.70*	84.70
3	FaiTH (DS-V4-Flash)	Cloud	81.70	86.49	69.86	79.53	*86.98*
4	FaiTH (Gemma4:31b-nvfp4)	Local 20 GB	80.52	84.40	70.92	78.40	85.68
5	FaiTH (Qwen3.6:27b-mtp-q8_0)	Local 29 GB	78.08	80.34	**72.48**	79.93	73.57
6	Temp-R1 (SOTA baseline)	–	78.00	88.80	55.00	71.40	96.90
7	FaiTH (Gemma4:12b-mlx-bf16)	Local 23 GB	77.18	80.59	68.75	77.41	76.62
8	FaiTH (Gemma4:26b-mxfp8)	Local 26 GB	73.12	76.11	65.74	74.01	70.96
9	FaiTH (Qwen3.6:27b-nvfp4)	Local 19 GB	69.81	69.06	71.65	81.22	42.00
10	FaiTH (GLM-4.7-Flash:q4_K_M)	Local 19 GB	69.29	72.60	61.09	74.56	56.45
11	FaiTH (GLM-4.7-Flash:q8_0)	Local 31 GB	67.70	69.78	62.56	75.68	48.26
12	FaiTH (Qwen3.6:35b-a3b-nvfp4)	Local 21 GB	64.37	65.02	62.77	74.27	40.24

Table 11: FaiTH on TL-CRON (full test set, Hits@1 %). Best and second-best FaiTH configuration per column are in **bold** and *italic*; Temp-R1 is listed for reference and is not ranked.

#	Backbone	Deploy	Overall	Simple	Medium	Complex
1	FaiTH (DS-V4-Flash)	Cloud	**71.88**	82.92	**59.03**	**74.66**
2	FaiTH (DS-V3.2)	Cloud	*70.67*	82.54	*57.80*	*72.85*
3	Temp-R1 (SOTA baseline)	–	70.50	96.00	48.60	67.20
4	FaiTH (GLM-4.7-Flash:q4_K_M)	Local 19 GB	70.23	82.88	56.67	72.41
5	FaiTH (Qwen3.6:27b-nvfp4)	Local 19 GB	69.82	**83.25**	55.69	71.91
6	FaiTH (Qwen3.6:27b-mtp-q8_0)	Local 29 GB	69.64	**83.25**	55.47	71.63
7	FaiTH (GLM-4.7-Flash:q8_0)	Local 31 GB	69.25	*83.21*	55.83	70.31
8	FaiTH (GPT-5.4)	Cloud	68.47	*83.21*	55.62	68.46
9	FaiTH (Gemma4:31b-nvfp4)	Local 20 GB	68.22	82.83	54.67	68.90
10	FaiTH (Qwen3.6:35b-a3b-nvfp4)	Local 21 GB	66.89	82.54	52.93	67.15
11	FaiTH (Gemma4:12b-mlx-bf16)	Local 23 GB	65.53	82.75	53.58	62.89
12	FaiTH (Gemma4:26b-mxfp8)	Local 26 GB	62.44	79.62	51.54	58.91

C.6 ROBUSTNESS TO TKG NOISE

We stress-test FaiTH (DS-V4-Flash) under two families of controlled corruption. *Execution-level noise* perturbs the fact store that Stage 3 operates on, via random fact insertion, timestamp shifting ($\pm$30 days), and fact deletion. *Grounding-level noise* perturbs the symbol space that Stage 2 resolves mentions against, via relation renaming (replacing canonical names with synonyms), entity-name reformatting (breaking the "Role (Country)" pattern), and their combination. Each type is applied at ratios of 5%, 10%, 15%, and 20%.

Tables 13 and 14 show that FaiTH degrades gradually under all noise types without catastrophic failure. Fact insertion has near-zero impact on both benchmarks (at most 0.84 pp at 20%), because

Table 12: FaiTH on TL-ICEWS (full test set, Hits@1 %). Best and second-best FaiTH configuration per column are in **bold** and *italic*; Temp-R1 is listed for reference and is not ranked.

#	Backbone	Deploy	Overall	Simple	Medium	Complex
1	FaiTH (Gemma4:31b-nvfp4)	Local 20 GB	**69.71**	**83.53**	53.25	**71.22**
2	FaiTH (DS-V3.2)	Cloud	*69.55*	83.40	**53.88**	69.70
3	FaiTH (DS-V4-Flash)	Cloud	69.43	82.98	53.28	*70.88*
4	FaiTH (Qwen3.6:27b-mtp-q8_0)	Local 29 GB	69.24	83.35	52.91	70.00
5	FaiTH (Qwen3.6:27b-nvfp4)	Local 19 GB	68.78	83.32	52.76	68.24
6	FaiTH (GPT-5.4)	Cloud	68.64	83.25	53.11	67.18
7	FaiTH (GLM-4.7-Flash:q4_K_M)	Local 19 GB	68.53	81.60	*53.41*	69.21
8	FaiTH (GLM-4.7-Flash:q8_0)	Local 31 GB	68.20	81.93	53.00	67.82
9	FaiTH (Qwen3.6:35b-a3b-nvfp4)	Local 21 GB	67.56	83.10	51.97	64.51
10	FaiTH (Gemma4:26b-mxfp8)	Local 26 GB	66.28	83.13	48.41	64.54
11	Temp-R1 (SOTA baseline)	–	64.20	86.60	38.80	59.50
12	FaiTH (Gemma4:12b-mlx-bf16)	Local 23 GB	61.54	*83.43*	40.45	55.84

randomly inserted facts almost never satisfy the conjunction of relation, role, and temporal constraints that the symbolic engine enforces. Timestamp corruption and fact deletion cause approximately linear degradation. Grounding-level noise is likewise absorbed: mixed noise at 20% retains 72.25% Hits@1 on MULTITQ. Entity corruption is milder than relation renaming as the multi-rule cascade of the entity grounder ϕ_E provides fallback paths.

Table 13: Noise robustness of FaiTH (DS-V4-Flash) on MULTITQ (Hits@1 %). Stage-3 (execution-level) noise perturbs the fact store and is scored on a 5,000-question subset against the full-run 81.93% baseline. Stage-2 (grounding-level) noise perturbs entity and relation naming conventions and is evaluated on the full 54,584-question test set. The two groups use separate clean baselines due to independent runs, where those baselines' metric variations are explained in Appendix C.2.

Target Stage	Noise Type	Questions	Clean	5%	10%	15%	20%
Stage 3 (Execution)	Fact insertion	5,000	81.93	81.72	81.72	81.68	81.60
Stage 3 (Execution)	Timestamp corruption	5,000	81.93	79.08	77.80	75.22	73.04
Stage 3 (Execution)	Fact deletion	5,000	81.93	78.60	76.26	73.82	71.24
Stage 2 (Grounding)	Relation renaming	54,584	81.64	80.36	77.90	77.84	76.59
Stage 2 (Grounding)	Entity corruption	54,584	81.64	80.88	79.82	78.99	76.99
Stage 2 (Grounding)	Mixed	54,584	81.64	79.61	76.18	75.35	72.25

Table 14: Noise robustness of FaiTH (DS-V4-Flash) on the full TL-ICEWS test set (Hits@1 %), with a clean baseline of 69.43%. Entity corruption perturbs Stage 2 (grounding), while fact insertion and deletion perturb the fact store of Stage 3 (execution).

Noise Type	5%	10%	15%	20%
Entity corruption	69.19	68.80	68.35	67.62
Fact insertion	69.15	69.00	68.80	68.59
Fact deletion	67.20	64.63	61.88	60.82

C.7 FINE-GRAINED ABLATION DECOMPOSITION

To investigate the ablation anomaly reported in Table 4, where removing temporal filtering appeared to increase the Time score, we decompose the ablation to the level of (question type $\times$ answer type) using DS-V4-Flash instead of the original DS-V3.2. Not all question types have both entity and time answers: only *equal* and *first_last* contain time-answer questions, while the remaining four types consist entirely of entity-answer questions. Table 15 resolves the anomaly. With DS-V4-Flash, the overall Time score drops from 85.72% to 84.59% under *w/o tf*, attributing that increase to a backbone-specific compilation artifact (Case C3 in Appendix C.4). Removing temporal filtering causes catastrophic failure on entity answers (80.37% to 15.65%). The *w/o ra* column degrades

every subtype, whereas *w/o ds* concentrates its damage on the compositional entity types, where *after_first* falls from 68.99% to 51.61% and *before_last* from 70.19% to 54.03%, and leaves the two time subtypes above the full pipeline.

Table 15: Fine-grained ablations of FaiTH (DS-V4-Flash, 81.93% run) on the full MULTITQ test set (Hits@1 %), by question type and answer type; only *equal* and *first_last* contain time answers.

Question type	Answer type	Full	w/o ra	w/o tf	w/o ds
equal	entity	92.96	91.97	0.59	92.93
equal	time	83.51	75.96	81.49	86.75
before_after	entity	82.43	75.85	31.11	81.70
first_last	entity	74.31	54.28	69.14	70.60
first_last	time	87.07	78.77	86.48	87.37
equal_multi	entity	72.19	62.36	21.55	69.85
after_first	entity	68.99	54.34	12.35	51.61
before_last	entity	70.19	54.14	10.42	54.03
Overall	entity	80.37	72.10	15.65	74.48
Overall	time	85.72	77.71	84.59	87.14
Overall	all	81.93	73.73	35.71	78.16

Table 16: Answer coverage of FaiTH (DS-V4-Flash) on MULTITQ (full test set, %). These statistics measure whether the gold answer appears in the Stage-3 evidence set and in the candidate list.

Question Type	In Evidence	In Candidates
equal	90.38	91.17
before_after	93.94	92.22
first_last	85.79	86.43
equal_multi	85.63	84.72
after_first	83.40	72.79
before_last	82.15	75.12
Overall	88.20	86.15

Table 17: Error attribution on MULTITQ (full test set, 54,584 questions) for the 81.93% run of FaiTH (DS-V4-Flash), which answers 9,864 questions incorrectly.

Error Category	Share
Retrieval miss (gold absent from evidence)	37.83%
Stage-4 error (gold in candidate list)	23.51%
Stage-4 error (gold in evidence only)	19.86%
Empty evidence after recovery	14.75%
Grounding failure	4.05%

C.8 ANSWER COVERAGE ANALYSIS

We measure the fraction of MULTITQ questions whose gold answer appears in the Stage-3 evidence set and in the extracted candidate list, using FaiTH (DS-V4-Flash). Table 16 shows an overall evidence coverage of 88.20%, substantially above the 71.78% top-50 coverage reported by PoK (Qian et al., 2025) for dense retrieval on the same benchmark. The gap between evidence and candidate coverage concentrates on the compositional types *after_first* and *before_last*, where the boundary extractor faces ordering ambiguity. On *equal* and *first_last*, the two types with time answers, candidate coverage is instead slightly higher than evidence coverage, because $\mathrm{Boundary}_\gamma$ truncates timestamps to the granularity γ of the specification and can emit a gold timestamp that no fact of $\mathcal{F}_q$ carries.

C.9 STAGE-WISE ERROR ATTRIBUTION

We trace all incorrectly answered MULTITQ questions of FaiTH (DS-V4-Flash) to the first pipeline stage whose output broke the chain to the gold answer. Table 17 shows that grounding failure

Table 18: Deterministic shortcut activation (% of questions resolved with one LLM call) on the full MULTITQ and TL-ICEWS test sets, where "agg." stands for aggregate. The breakdown comes from a separate full run, so its overall rates differ slightly from those in Table 7.

MULTITQ		
Question Type	1 Call	2 Calls
equal	61.00	39.00
before_after	24.76	75.24
first_last	90.98	9.02
equal_multi	58.47	41.53
after_first	97.13	2.87
before_last	95.81	4.19
Single (agg.)	58.76	41.24
Multiple (agg.)	88.72	11.28
Overall	67.39	32.61
TL-ICEWS		
Difficulty	1 Call	2 Calls
Simple	79.25	20.75
Medium	64.24	35.76
Complex	66.42	33.58
Overall	70.88	29.12

Table 19: Multi-metric evaluation of FaiTH (DS-V4-Flash, 81.64% run) on the full MULTITQ test set (%) by question type. The three LLM-Judge columns use DS-V4-Flash, i.e., the answering model itself, and two OpenAI models, GPT-6-Luna and GPT-6-Sol, as judges.

				LLM-Judge		
Category	Hits@1	EM	F1	DS-V4-Flash	GPT-6-Luna	GPT-6-Sol
equal	89.38	82.61	85.58	89.40	89.32	89.31
before_after	82.22	73.88	77.44	82.75	82.38	82.04
first_last	85.80	85.78	85.83	85.83	85.82	85.80
equal_multi	71.50	69.07	70.95	71.72	71.31	71.25
after_first	68.83	68.72	69.38	69.07	68.90	68.90
before_last	70.24	69.94	70.55	70.24	70.21	70.10
Overall	81.64	77.61	79.54	81.80	81.65	81.56

Table 20: Multi-metric evaluation of FaiTH (DS-V4-Flash) on the full TL-ICEWS test set (%) by answer type, for the same run as Table 2.

				LLM-Judge		
Answer type	Hits@1	EM	F1	DS-V4-Flash	GPT-6-Luna	GPT-6-Sol
duration	91.37	91.37	94.03	91.37	91.37	91.37
object	63.18	62.82	68.82	64.20	63.04	63.00
subject	44.62	43.83	44.79	46.04	44.10	43.86
timestamp_start	91.06	91.06	91.06	91.06	91.06	91.06
timestamp_end	92.10	92.10	92.10	92.30	92.10	92.10
timestamp_range	90.84	90.84	93.66	90.84	90.84	90.84
relation_duration	79.90	79.90	85.90	84.91	84.57	84.57
relation_ranking	95.12	95.12	95.12	95.12	95.12	95.12
relation_u/i	61.11	61.11	68.30	91.66	91.30	91.30
Overall	69.43	69.20	73.00	76.71	75.98	75.93

accounts for only 4.05% of the 9,864 errors, that is, about 400 questions or 0.73% of the benchmark, so grounding costs FaiTH less than one point of Hits@1. The largest error (37.83%) involves retrieval misses under a correctly grounded specification, dominated by Stage-1 compilation defects.

Table 21: Handcrafted (Hand) vs. LLM-generated (LLM) grounding rules (FaiTH, DS-V4-Flash, full test set on MULTITQ, %). Within each benchmark all metrics come from one run, so the comparison is fair. "r" for "role-title" and "c" for "country/demonym".

Benchmark	Grounder	Rules	Hits@1	EM	F1
MULTITQ	Handcrafted	84 r + 101 c	81.42	77.19	79.15
	LLM-generated	547 r + 326 c	80.01	75.79	77.76

C.10 DETERMINISTIC SHORTCUT STATISTICS

Table 18 breaks down the LLM call consumption by question type on both benchmarks. On MULTITQ a counter-intuitive finding emerges, namely that compositional questions activate the shortcut far more often than simple ones: Multiple-type questions use only one call 88.72% of the time against 58.76% for Single, because *after_first* (97.13%) and *before_last* (95.81%) almost always resolve to a single boundary entity after temporal sorting. The lowest activation is *before_after* (24.76%), whose open-ended windows leave multiple valid candidates requiring LLM disambiguation.

C.11 MULTI-METRIC EVALUATION

Hits@1 alone can mask formatting effects, therefore we additionally score FaiTH (DS-V4-Flash) with three complementary metrics. First, *Exact match* (EM) requires the normalized prediction to equal the normalized gold answer, with a tolerance of at most three characters of substring difference. Second, *Token-level F1* computes precision and recall over whitespace-delimited tokens, awarding partial credit. Third, *LLM-as-judge* prompts an LLM to decide semantic equivalence, crediting correct answers expressed in alternative surface forms; we report three judges, namely the answering model DS-V4-Flash itself and two models from a different vendor, GPT-6-Luna and GPT-6-Sol (OpenAI, 2026), with thinking mode disabled for all three LLMs. The expected ordering is Hits@1 $\geq$ EM and F1 $\geq$ EM, with LLM-Judge at or above Hits@1. Tables 19 and 20 confirm it for both string metrics and for the DS-V4-Flash Judge; F1 can also exceed Hits@1, because partial token overlap earns credit where exact matching gives none, and the two slightly stricter OpenAI judges fall at most 0.76 pp below Hits@1 on a few rows, because substring matching credits any prediction that contains the gold string while the judges require the same entity or timestamp. The three judges agree within 2.18 pp on every row and within 0.78 pp overall, so the LLM-Judge scores do not hinge on the answering model judging its own outputs. On MULTITQ, all metrics span only 4.19 pp (from EM at 77.61% to the DS-V4-Flash Judge at 81.80%), indicating clean output formatting. On TL-ICEWS, all three Judge scores on *relation_union_or_intersection* (91.30% to 91.66%) far exceed the Hits@1 of 61.11%, because many interval answers are semantically correct but rendered in a different surface form that string metrics fail to match.

C.12 LLM-AUTOMATED GROUNDING

The handcrafted synonym tables are the main per-TKG engineering cost of FaiTH, so we test whether an LLM (DS-V4-Flash) can generate them automatically from the TKG vocabulary alone, without access to any question. The LLM-based generation proceeds in three phases: role-title paraphrases (two rounds, with the second targeting uncovered prefixes), country and demonym mappings, and rule validation against the TKG to discard non-existent targets. The generated tables replace the handcrafted ones while all other pipeline components remain unchanged. Table 21 shows that the LLM-automated grounder costs only 1.41 pp on MULTITQ (81.42% to 80.01%). The LLM generates far more rules than the handcrafted tables on MULTITQ, the largest TKGQA benchmark, yet the gap stays small because grounding accounts for only 4.05% of errors as discussed in Appendix C.9.

D SELECTED PROMPT TEMPLATES

We present selected prompt templates used by FaiTH: the Stage-1 compile call and the Stage-4 reason call. We show templates for two representative benchmarks: MULTITQ (point-valued timestamps) and TIMELINEKGQA-ICEWS (interval-valued facts with sentinel endpoints). For brevity, we show two or three representative few-shot examples per template.

D.1 Stage 1 Compile Prompt for MultiTQ

The following prompt is sent as a single user message. The LLM is instructed to return only a JSON object. The optional relation hint block is included when the TKG relation vocabulary is available.

Stage 1 Compile Prompt (MULTITQ)

```
You are an expert at analyzing temporal knowledge
graph questions.
Given ONLY a question, determine its type and extract
structured info as JSON.

Question types:
- equal: events at a specific time
- before_after: events before/after a time or entity event
- first_last: first or last occurrence
- equal_multi: time constraint + first/last, or same
- after_first: first event after a reference
- before_last: last event before a reference

Output JSON schema:
{
  "question_type": "equal | before_after | first_last
    | equal_multi | after_first | before_last",
  "anchor_entity": "main entity being queried about",
  "relation_mention": "the action or event described",
  "anchor_role": "subject | object",
  "ref_entity": "entity as temporal reference, or null",
  "ref_time": "explicit time in ISO 8601, or null",
  "time_filter": "equal | before | after | null",
  "sort_order": "asc | desc | null",
  "answer_type": "entity | time",
  "time_granularity": "day | month | year"
}
Rules:
- anchor_role: "object" if anchor receives the action,
  "subject" if anchor performs the action
- Normalize dates to ISO 8601
- Output ONLY the JSON object.

Available KG relations:
- Accuse
- Arrest, detain, or charge with legal action
- Cooperate economically
  ... (248 more relations omitted)

Examples:
Q: Which country provided humanitarian aid to Sudan
   in 2007?
A: {"question_type":"equal","anchor_entity":"Sudan",
"relation_mention":"provided humanitarian aid",
"anchor_role":"object","ref_entity":null,
"ref_time":"2007","time_filter":"equal",
"sort_order":null,"answer_type":"entity",
"time_granularity":"year"}

Q: In Dec, 2008, who would wish to negotiate with
   the Senate of Romania?
A: {"question_type":"equal","anchor_entity":
"Senate of Romania","relation_mention":
"wish to negotiate","anchor_role":"object",
"ref_entity":null,"ref_time":"2008-12",
"time_filter":"equal","sort_order":null,
"answer_type":"entity","time_granularity":"month"}

Q: After November 2007, who wanted to engage in
   diplomatic cooperation with Timor-Leste?
A: {"question_type":"before_after",
"anchor_entity":"Timor-Leste","relation_mention":
"engage in diplomatic cooperation",
"anchor_role":"object","ref_entity":null,
"ref_time":"2007-11","time_filter":"after",
"sort_order":null,"answer_type":"entity",
"time_granularity":"month"}

  ... (9 more examples omitted)

Q: <input question>
A:
```

D.2 Stage 4 Reason Prompt for MultiTQ

This prompt is issued when the deterministic shortcut of Eq. (14) does not fire. The evidence block contains the chronologically sorted facts from the faithful evidence set $\mathcal{F}_q$ produced by Stage 3. The metadata fields `qtype`, `tf`, `so`, and `answer_type` are populated from $\hat{\mathcal{C}}_q$.

Stage 4 Reason Prompt (MultiTQ)

```
You are a factual QA system. Evidence below comes from
an authentic temporal knowledge graph.

Rules:
1. Provide the best answer based on the evidence.
   Prefer giving a concrete answer over expressing
   uncertainty.
2. For entity answers: use the EXACT full entity name
   as shown in evidence, including parenthetical
   qualifiers.
   Example: "Children (Thailand)" not just "Children".
3. For time answers: use format <fmt>.
4. If the question asks "which country", prefer
   country-level entities over sub-entities.
5. The answer must come from entities/times present
   in the evidence.

Context:
- Question type: <qtype>,
  Constraints: time_filter=<tf>, sort_order=<so>
- Answer type: <answer_type>

Evidence (sorted by time):
1. Head of Government (France) | Make a visit |
   Japan | 2010-03-01
2. Citizen (Thailand) | Make a visit |
   Japan | 2010-05-22
3. Head of State (China) | Make a visit |
   Japan | 2010-11-13
  ... (remaining evidence facts)

Question: <input question>

Answer:
```

D.3 Stage 1 Compile Prompt for TL-ICEWS

The TL-ICEWS compile prompt differs from MultiTQ in three aspects: (i) every event template uses the fixed predicate `Affiliation To`; (ii) the answer slot inventory is expanded to include dur, rdur, rank, tsr, and setop; and (iii) the `ref_relation` field also supports `finishes_same_time` and `starts_same_time` besides standard predicates.

Stage 1 Compile Prompt (TimelineKGQA-ICEWS)

```
You are an expert parser for ICEWS Actor affiliation
questions.
Every fact uses the same predicate 'Affiliation To'.

Output a JSON spec with this exact schema:
{
  "events": [
    {"subject": "<name or ?>",
     "predicate": "Affiliation To",
     "object": "<name or ?>"}
  ],
  "answer_slot": "subject | object | timestamp_start
    | timestamp_end | timestamp_range | duration
    | relation_duration | relation_ranking
    | relation_union_or_intersection",
  "time_constraint": {"op": "during|before|after|equal",
    "start": "YYYY-MM-DD", "end": "YYYY-MM-DD"} or null,
  "temporal_op": "sum | average | compare | intersection
    | intersection_duration | union | ranking_start
    | ranking_end | ranking_duration" or null,
  "ref_relation": "before | after | during | same_time
    | finishes_same_time | starts_same_time | none",
  "ranking_target": "<entity whose rank is asked>"
```

```
    or null
}

Rules:
1. Always put "Affiliation To" as the predicate.
2. Put "?" in the slot the question is asking about.
3. Number of events = number of distinct triples.
4. CAREFUL with passive voice: "X Affiliation To by who"
   means the UNKNOWN is the SUBJECT.
5. "who finishes ..." asks WHO (a subject).
   Set answer_slot to "subject".
6. Time fields may literally be "beginning of time" or
   "end of time". Keep them verbatim.
7. ref_relation: only set when >=2 events are connected
   by a temporal phrase.
8. Output ONLY the JSON object.

Examples:
Q: Who Affiliation To Ministry of Defense from
   2014-02-22 to end of time?
A: {"events":[{"subject":"?",
"predicate":"Affiliation To",
"object":"Ministry of Defense"}],
"answer_slot":"subject",
"time_constraint":{"op":"during",
"start":"2014-02-22","end":"end of time"},
"temporal_op":null,"ref_relation":"none",
"ranking_target":null}

Q: Vyacheslav Kyrylenko Affiliation To which
   organisation from 2006-12-01 to 2007-03-30?
A: {"events":[{"subject":"Vyacheslav Kyrylenko",
"predicate":"Affiliation To","object":"?"}],
"answer_slot":"object",
"time_constraint":{"op":"during",
"start":"2006-12-01","end":"2007-03-30"},
"temporal_op":null,"ref_relation":"none",
"ranking_target":null}

  ... (5 more examples omitted)

Q: <input question>
A:
```

D.4 Stage 4 Reason Prompt for TL-ICEWS

The TL-ICEWS reason prompt extends the MULTITQ variant with interval-valued evidence and an explicit instruction to handle sentinel endpoints.

Stage 4 Reason Prompt (TIMELINEKGQA-ICEWS)

```
You are answering a question over a temporal knowledge
graph.
Each fact is:
  subject | predicate | object | start_time -> end_time.
Time fields may be the literal strings
'beginning of time' or 'end of time'.

Rules:
1. Preserve entity names EXACTLY as shown in the
   evidence.
2. Respect all temporal constraints in the question.
3. Output ONLY the final answer.
4. For time answers copy the value verbatim.
5. The answer MUST appear in the evidence.

Evidence (sorted by time):
  1. Atta Mohammed Nur | Affiliation To |
     Northern Alliance | 2001-11-01 -> 2014-09-29
    ... (remaining evidence facts)

Question: <input question>
Expected answer kind: <slot>
Answer:
```