Scaffolding Matters More Than the Schema: A Design and Case Study Report on Targum, a Controlled-Vocabulary Translation Engine for Esoteric Primary Sources

Author: Vincent W. Couey Independent Researcher · Hekhal Project · Toledo, Ohio · vinnycouey@gmail.com · hekhal.org/targum-experiments/scaffolding-matters-paper **Status:** Submitted draft v3, 2026-05-16. Target venue: Aries: Journal for the Study of Western Esotericism (Brill); secondary candidate Digital Humanities Quarterly. **Companion artifacts:** Benchmark specifications, run audit packages (assembled prompts, validated outputs, drift / registry / citation reports, adversarial-review transcript) and the verified-citation manifest excerpt are published as a public methodology reproducibility package at hekhal.org/targum-experiments/scaffolding-matters-paper. The Targum engine source, the controlled glossaries, the frame controllers, and the curated scholarly-corpus seeds are proprietary editorial work of the Hekhal Project and are not released publicly; the engine code is available to the academic peer-review process under standard reviewer-disclosure terms by request to the corresponding author.

Abstract

We describe Targum, a controlled-vocabulary translation engine for the esoteric and contemplative primary-source corpus, and report case-study findings from three benchmark passages across two scripts and three traditions. Targum is a layered pipeline (reference resolution, multilingual morphology, hard-constraint glossary, retrieval over a curated scholarly corpus, hermeneutic frame controllers, schema-validated generation, drift audit, registry check, citation verification, and editor sign-off) built over the editorial infrastructure of the Hekhal cross-tradition reference (hekhal.org). The architectural design separates the engine (deterministic pipeline) from the scaffolding (per-corpus glossaries, frame controllers, scholarly summaries, and lexicon registries that the engine consumes at generation time).

Across three case-study benchmarks (Ibn Arabi Tarjuman al-Ashwaq XI.13–15; the kuntu kanzan Akbarian hadith; Pseudo-Dionysius Mystical Theology I.1) we report two findings.

First, when the scaffolding for a passage's corpus is built (controlled glossary at the per-corpus calibration bar; frame controller with named interpretive relations; lexicon pages on the public site for cross-references to resolve), the engine produces output whose distinctive value over standard public-domain translations is locatable in interpretive surface area — ambiguity preservation, frame-relation footnotes, glossary-update proposals, attribution-problem disclosure — that the registry check verifies resolves to extant infrastructure. Independent adversarial review of one scaffolded output (Pseudo-Dionysius MT I.1, post-scaffolding) by a cold-context reviewer with no knowledge of the experimental hypothesis judged it "editor-grade output… suitable for editor sign-off" with two minor compliance fixes.

Second, when the scaffolding is partial or absent, the engine produces schema-valid output that references infrastructure that does not exist on disk. The original adversarial Test C, run before the kataphatic-apophatic frame controller and Christian-apophatic glossary were built, emits a lexicon reference (`lexicon/koryphe`) and a cross-reference (`dionysius.mystical-theology.1.3`) that no Hekhal page implements. Importantly, several other references in the same run that we initially characterized as fabrications resolve correctly when measured

against the current Hekhal lexicon — they reference pages that have been authored in the interval between the original run and this paper. The honest finding is therefore narrower than the strongest framing: schema-valid output can reference nonexistent infrastructure, but the magnitude of this failure mode is corpus-and-time-dependent, and post-LLM registry checks against the current infrastructure catch it.

We argue for a design discipline we call registry-grounded translation: every schema-shape commitment must be paired with a post-LLM existence check against the actual infrastructure (frame controllers on disk, lexicon pages on the public site, glossary revisions in version control, citations in a verified manifest). We publish the benchmark specifications, the full run audit packages (assembled prompts, validated outputs, drift / registry / citation reports), the adversarial-review transcript, and a verified-citation manifest excerpt as a public reproducibility surface; the engine source code and the corpus-specific scaffolding remain proprietary editorial work of the Hekhal Project and are available to the journal's peer-review process under standard reviewer-disclosure terms.

Keywords: machine translation, digital humanities, esoteric studies, controlled vocabulary, retrieval-augmented generation, Pseudo-Dionysius, Ibn Arabi, large language models, hermeneutic theory.

Key contributions

1. A controlled-vocabulary translation engine, Targum, layered over the editorial infrastructure of an active cross-tradition esoteric reference. The architecture is described in §3 at the level required to replicate the design in an independent implementation.
2. The articulation and empirical defense of a design discipline we call registry-grounded translation: schema-shape commitments must be paired with post-LLM existence checks against the actual infrastructure on disk (frame controllers, lexicon pages, glossary revisions, citation manifest). The discipline is formalized as two new pipeline layers — Layer 6.5 (registry check) and Layer 6.6 (citation verification) — and shown in §6 to catch a class of failure (vacuous compliance) that schema enforcement alone does not.
3. Three case-study benchmarks across two scripts and three traditions (Akbarian Sufi poetry, Akbarian doctrinal hadith, Christian apophatic prose), with per-run audit packages preserving every artifact necessary to inspect-verify the empirical claims. One scaffolded output judged "editor-grade output … suitable for editor sign-off" by an independent cold-context reviewer.
4. A practical contribution: the demonstration that the post-LLM verification layer reframes the open-source presumption that has dominated digital-humanities methodology work. Reviewer-verifiable claims do not require open-source engine code; they require open audit packages. The audit packages for this paper are publicly available at the URL named in §9.1.

---

1. The gap Targum fills

The academic study of Western esotericism as a distinct disciplinary field is a comparatively recent consolidation (Faivre 1994; Hanegraaff 2012; Versluis 2007), and the available open public-internet infrastructure for the field's primary sources has not kept pace with the discipline's growth. Most foundational esoteric primary sources sit in one of three places online: Wikipedia-grade summary;

paywalled academic editions behind Brill, JSTOR, or comparable academic-press gates; or modern-occult and new-age material whose relationship to the primary tradition ranges from creative reception to outright fabrication. The open public-internet ecosystem for serious cross-tradition esoteric study is, in 2026, materially thinner than the comparable ecosystem for biblical studies (which has Sefaria, Perseus, and a robust open critical apparatus), patristic studies (CCEL, New Advent), or even Islamic scriptural studies (Quran.com, al-Islam.org, Sunnah.com). The Hekhal project, of which Targum is the translation arm, exists to close that gap.

The long-tail problem is concrete. Take any Tier-1 mystical corpus and inventory what is available in open English: nearly all foundational pre-1929 translations exist on the Internet Archive and sacred-texts.com (Nicholson on the Tarjuman; Parker, Rolt, and Luibhéid on the Areopagitic corpus; Mead on the Corpus Hermeticum; Westcott on the Sefer Yetzirah; Mathers on the Kabbalah Unveiled; Sparrow on Boehme; Underhill on the Cloud of Unknowing; Inge and Pfeiffer on Eckhart; MacKenna on Plotinus). What does not exist in open English, in any rigorous form, is roughly:

- Most of the Akbarian secondary corpus (Qashani, Jandi, Qaysari)
- The Lurianic Kabbalistic corpus past the few passages in Scholem and Idel
- Most of the Hesychast literature past the Philokalia anthology
- Nearly all of Ismaili esoteric exegesis (Sijistani, Kirmani)
- Hadith collections (canonical and contested) bearing Sufi or contemplative weight — including the kuntu kanzan (Hidden Treasure) saying that is the doctrinal hinge of Akbarian metaphysics yet has no PD English

The pre-1929 translation tradition is unrepublishable as a comprehensive coverage strategy because its scope is what it is: Victorian and Edwardian scholars translated what their time and theological commitments selected. Modern critical editions (Pritzker Zohar, Hayman Sefer Yetzirah, Luibhéid Pseudo-Dionysius) are copyrighted and unredistributable. The legitimate paths to comprehensive open coverage are: reuse PD translations where they exist; reproduce CC-licensed corpora where they exist (Sefaria); host original-language ancient texts which are always PD; commission new translations; and produce AI-assisted fresh translations from PD originals, released under the project's license. This last path is what Targum is built to industrialize.

The risk in AI-assisted translation of esoteric primary sources is well known. A generic large language model prompted to translate produces fluent English that systematically over-domesticates technical vocabulary, flattens hermeneutic distinctions specific to the corpus, hallucinates apparatus material that resembles scholarly footnotes without grounding in actual scholarship, and produces output that cannot be audited because there is no machine-readable trace of which interpretive choices were made and why. Targum is built around the claim that these failure modes are not inevitable properties of LLM translation but artifacts of unscaffolded prompting; that a purpose-built engine with controlled lexicon, hermeneutic frame controllers, retrieval-augmented scholarly context, a strict output schema, and post-LLM verification against the actual infrastructure on disk can produce output whose interpretive choices are interrogable, whose vocabulary is controlled, and whose claims trace to verifiable infrastructure.

This paper reports the case-study results of testing that claim, with explicit attention to the failure modes the design surfaces.

A note on the name. Targum (plural Targumim) is a class of Aramaic translations of the Hebrew Bible; we use it as a proper-noun project name deliberately, as an act of homage and to signal the project's tradition-respecting editorial stance, not in ignorance of the term's existing referents. The Hekhal Project's editorial conventions name texts by their primary tradition's term and reserve

internal jargon for engineering surfaces.

---

2. Related work

Targum sits at the intersection of four research lines.

Digital editions of primary religious corpora. Sefaria (sefaria.org) is the canonical model: open Hebrew/Aramaic corpus with linked English translations, Creative Commons licensed where the translation permits, with a parallel-reader interface and structured citation infrastructure. Perseus Digital Library does the analogous work for classical Greek and Latin. OpenITI (openiti.org) hosts the canonical text of much of the medieval Arabic Islamic textual corpus in normalized form. None of these is a translation engine; they index existing texts and existing translations.

Generic LLM translation of low-resource and contested-register text. The recent literature on prompting LLMs for translation has demonstrated (Vilar et al. 2023; Hendy et al. 2023; Garcia et al. 2023) that competitive translation quality is achievable for many language pairs with appropriate prompting. The mystical-register and esoteric-vocabulary case has received less attention; ad-hoc prompting consistently produces over-domestication and apparatus fabrication, but no purpose-built engine has been published.

Retrieval-augmented generation for technical domains. RAG architectures (Lewis et al. 2020; Gao et al. 2024) layer a retrieval step over LLM generation so the model has access to authoritative reference material at generation time. Targum's retrieval layer implements a hybrid BM25 plus multilingual-E5 dense retriever (Wang et al. 2024) over a curated scholarly corpus.

Controlled-vocabulary and constrained-decoding approaches in technical translation. The medical, legal, and scientific-translation literatures have developed controlled-vocabulary approaches (Schäffner 2002; Kockaert and Steurs 2015) where domain-specific terminology databases gate translation choices. Most existing implementations rely on pre-LLM rule-based systems; LLM-based translation engines have largely abandoned controlled-vocabulary discipline in favor of fluency. Targum's design returns controlled vocabulary to LLM translation via prompt-time hard-constraint injection and post-LLM drift audit.

Targum's novelty over these four lines is the combination plus the post-LLM verification layer: a translation engine that takes a controlled lexicon and hermeneutic frame controllers as first-class inputs alongside the retrieval-augmented context, produces output validated against a strict semantic schema, emits an audit trail that traces every interpretive choice, and runs registry and citation checks against the actual infrastructure on disk before the output is offered to an editor for sign-off. The corpus-specific scaffolding (glossary, frame controller, scholarly corpus, lexicon pages, verified-citation manifest) is what distinguishes interrogable controlled-vocabulary output from merely fluent unscaffolded output.

---

3. Architecture

Targum is a layered pipeline. Each layer has explicit input and output contracts and a documented adopt-vs-build decision named in the master specification. The architecture is the substrate; the scaffolding is what makes a given corpus translatable through it.

Layer 0 — Reference resolution. Maps a canonical reference (e.g. `openiti:0638IbnArabi.RisalaAlAha` `ara1`) to a `Source` dict carrying the verbatim original-language text plus provenance metadata. Adopters: OpenITI canonical IDs, Sefaria refs, Perseus TLG IDs, Hekhal-internal slugs.

Layer 1 — Multilingual morphology. Tokenizes the source and emits a per-token morphological analysis. Adapters route by language: CAMeL Tools for Arabic and Persian; Dicta for Hebrew and Aramaic; CLTK for Greek, Latin, and Sanskrit; Stanza fallback for everything else. A naive whitespace fallback ensures the pipeline runs even on a clean install without heavy NLP dependencies.

Layer 2 — Glossary and frame controllers. Two parallel modules.

The glossary is a per-corpus controlled vocabulary stored as YAML, with each entry specifying a `selected_sense` (default rendering), `active_senses` (alternative renderings), `forbidden_renderings` (anti-patterns flagged by the drift audit), and an editorial note. The glossary is injected into the prompt as a hard-constraint block; the drift audit verifies post-hoc that the engine's output respects the constraints.

The frame controllers are per-tradition hermeneutic grammars (PaRDeS for Kabbalah; zahir/batin for Sufism; kataphatic/apophatic for Christian apophatic theology; the Quadriga for Western Christian exegesis; tafsir/ta'wil for Quranic exegesis; peshat/derash for rabbinic exegesis). Each frame controller is a YAML module specifying a prompt preface, rendering rules, refusal clauses, named interpretive relations, and a `senses_index`. Multiple frame controllers can co-activate when the source corpus permits (Pseudo-Dionysius activates both kataphatic-apophatic and Quadriga because the Western reception reads him through both).

Layer 3 — Retrieval over the scholarly corpus. The corpus is a per-collection set of editorial summaries authored by Hekhal Editorial. The retriever is hybrid BM25 plus dense (multilingual-E5) with Reciprocal Rank Fusion. BM25 is the default; dense is opt-in due to platform-specific fragility on the E5 safetensors mmap on Windows page files.

Layer 4 — Prompt assembly and generation. The orchestrator assembles a system message (role, frame controllers, glossary constraints, retrieved scholarly context, morph stream, output schema instructions) and a user message (chunk metadata plus verbatim source). The generation client (Anthropic Claude default; OpenAI and open-model adapters scoped; mock client for offline CI; and an in-session-LLM workflow described in §9.2) returns a parsed JSON dictionary.

Layer 4.5 — Cross-translator memory. Public-domain renderings of indexed passages are surfaced in the prompt as a prior-art block ("the engine compares against Westcott / Mead / Mathers / Nicholson / Parker / Underhill / MacKenna at the corresponding passage") and re-derived deterministically post-generation by a token-Jaccard scorer that maps engine-vs-translator overlap into agreement labels (full / substantial / partial / divergent / incomparable). Editor override remains canonical at review time.

Layer 5 — Schema validation. The output is validated against `backend/schema/translation_output.j` (JSON Schema draft 2020-12). Validation failure regenerates up to twice, then surfaces the structural errors for editor inspection.

Layer 6 — Drift audit. Verifies that the engine's output respects the glossary's `forbidden_renderings`, the frame-controller refusal clauses, and the glossary's `selected_sense` consistency. Incidents are surfaced for editor review.

Layer 6.5 — Registry check. A new Phase-1 commitment introduced in the methodology-paper

development pass and shipped alongside this report. Verifies that every `frame_controllers_applied[]` entry corresponds to a YAML on disk with matching `frame_id`; every `range_cards[].lexicon_ref` of shape `lexicon/{slug}` resolves to either a glossary term or a Hekhal lexicon page; every `apparatus.cross_references[].target_ref` resolves to a Hekhal page of the appropriate kind (lexicon, text, figure) unless its relation contains "proposed"; and every `audit_trail.lexicon_revision` matches an actual on-disk glossary revision. Incidents are surfaced; they do not block emit but they are required for the `verified` status flip in the editor workflow.

Layer 6.6 — Citation verification. A second new Phase-1 commitment. The engine carries a `corpus/verified_citations.yaml` manifest distinguishing citations that have been editor-verified against printed or scanned sources from citations reproduced from training memory and citations not yet appearing in the manifest at all. Every output citation is checked against the manifest and labelled `verified`, `training-memory`, or `unverified`. The discipline is humble — we do not auto-fetch from Google Scholar or WorldCat in this version — but the manifest provides a single editorial surface where verification work accumulates rather than being repeated per output.

Layer 7 — Editor workflow. Translations carry a seven-state `TranslationStatus` taxonomy (verified / public-domain / machine-assisted / translation-pending / community / commissioned / gold), gated at the editor sign-off step; nothing auto-publishes. The status flip from `machine-assisted` to `verified` requires that drift audit, registry check, and citation check are all clean or whose remaining incidents have been individually resolved or accepted by the editor.

---

4. Phase-1 architectural commitments

The architecture above is the substrate. Seven specific commitments distinguish Targum from prompting a generic LLM:

1. **Controlled lexicon as first-class input.** Every term in Hekhal's lexicon is retrieved per passage; consistency becomes mechanical rather than heroic. Currently shipped at 49 entries across 4 glossary files (Akbarian general at 20 entries; Akbarian-specific at 1; Christian-apophatic at 15; Kabbalah at 13).
2. **School and period scoping.** A Bahir passage is translated against Provençal-Catalan vocabulary, not Lurianic; an early Akbarian passage against Andalusian, not Persian-Akbarian reception. Per-corpus glossary directories and the `source.school` / `source.period` fields the orchestrator passes into prompt assembly. All six current-generation frame controllers loaded.
3. **Hermeneutic-frame awareness.** Six frame controllers (PaRDeS, zahir-batin, kataphatic-apophatic, Quadriga, tafsir-ta'wil, peshat-derash) select per-corpus and inject prompt rules, refusal clauses, and named interpretive relations. The frame controller asks the apparatus footnote to surface the active relation between affirmation and negation rather than leaving the doctrinal content implicit.
4. **Cross-translator memory.** Layer 4.5 above. Currently shipped at 15 chunks / 28 PD renderings / 10 source files. The MVP deterministic scorer is token-Jaccard v1 with three thresholds (≥ 0.70 full, 0.50–0.70 substantial, 0.30–0.50 partial, below divergent).

5. **Letter-mystical phenomena detection.** Per-script gematria value tables (Hebrew with sofit alternate; Arabic abjad in the Mashriqi ordering; Greek isopsephy); rule-based detection over `source.original_text`; deterministic post-LLM stamping.
6. **Registry-grounded references.** Layer 6.5 above. Post-LLM verification that every infrastructure reference resolves. The Phase-1 commitment with the highest leverage on the failure mode discussed in §6 below.
7. **Verified-citation manifest.** Layer 6.6 above. Citations are not signed-off into `verified` status until they have been individually editor-checked against printed or scanned sources, and the verification work accumulates in a single manifest rather than being repeated per output.

---

5. Benchmark design

We selected three passages to stress-test the engine along three independent axes.

Test A — domain canonical. Ibn Arabi, Tarjuman al-Ashwaq, Poem XI lines 13–15 ("religion of love"). The most-cited Sufi-poetry passage in Akbarian Studies; the comparison set is rich (Nicholson 1911; Sells 1994; Chittick 1989); the glossary is at its calibration bar; the frame controller (zahir/batin) is fully implemented; the corpus is the engine's best-tuned. This is the test of what the engine produces when everything is in its favor.

Test B — domain doctrinal prose. The kuntu kanzan hadith (Hidden Treasure). Doctrinally foundational for Akbarian metaphysics (Ibn Arabi cites it as the substrate of his treatment of tajalli) yet contested as hadith: it does not appear in the canonical Sunni collections, is classified as "no transmissional origin" by al-Sakhawi and al-Ajluni, and "fabricated" by al-Albani. Tests whether the engine surfaces the attribution problem rather than treating the saying as canonical scripture, and whether it handles short doctrinal prose with the same interpretive discipline it brings to poetry.

Test C — adversarial out-of-domain (then re-run scaffolded). Pseudo-Dionysius, Mystical Theology I.1 prayer ("Trinity beyond being…"). At the time of the original benchmark run, the engine had no kataphatic-apophatic frame controller, no Christian-apophatic glossary, and no Christian-apophatic scholarly corpus. The test was deliberately adversarial: what does the engine produce when none of its scaffolding fires? The point was to establish the floor of what schema-enforced LLM generation alone produces. By the time of this paper's drafting, the requisite scaffolding has been built (the kataphatic-apophatic frame controller is shipped at 289 lines of YAML and validates clean; the Christian-apophatic glossary ships at 15 controlled entries). We re-ran Test C with the scaffolding firing and present both runs side-by-side.

For each test we fix the source language and pre-commit the source text to the spec YAML before any generation call. Reference translations are consulted only after the output JSON is finalized. PD references are reproduced verbatim where verified against archive.org or CCEL scans; copyrighted modern translations are characterized rather than reproduced. The drift audit, registry check, and citation check run; incidents are surfaced. Every artifact (spec, prompt, output, drift report, registry report, citation report) is preserved at `benchmarks/{id}/run-{timestamp}/`.

---

6. Results

6.1 The four-layer measurement framework

Every benchmark run reported here was measured against four independent layers of post-LLM verification:

1. **Schema validation** — JSON Schema draft 2020-12 enforcement against `back-end/schema/translation_output.json`.
2. **Drift audit** — `forbidden_renderings` violations and surface-form `selected_sense` mismatches against the glossary.
3. **Registry check** (new in Phase 1) — every infrastructure reference must resolve.
4. **Citation check** (new in Phase 1) — every apparatus citation must be in the verified-citation manifest as `verified` or explicitly marked `training-memory`.

Independent adversarial review by a cold-context auditor (Anthropic Claude in a fresh session, with no knowledge of the experimental hypothesis or the desired finding) was conducted for Test C scaffolded as an additional verification layer.

6.2 Test A — Tarjuman al-Ashwaq XI.13–15

The engine's primary rendering:

> Indeed my heart has become receptive to every form: a pasture for gazelles and a convent for monks, a temple for idols and a Ka'ba for the circumambulator, the tablets of Torah and the codex of Qur'an. I follow the religion of love wherever its mounts may turn; love is my religion and my faith.

The translation is competitive with Nicholson's 1911 PD English. The substantive value-add over Nicholson is locatable in three places.

Ambiguity preservation. The engine surfaces two doctrinal pivots that Nicholson's translation resolves silently. First, the qabilan polysemy: the unpointed Arabic licenses both "receptive to" (passive valence; the qalb-as-mirror doctrine in which the perfected heart is the surface on which every divine self-disclosure is registered) and "capable of" (active valence; the qalb/taqallub etymological pun). Nicholson selects the active reading silently; Targum's apparatus surfaces both with frame-relation notes mapping each to zahir and batin respectively. Second, the d-y-n root play in adinu bi-dini l-hubbi: the verb can read as "I follow the religion of love" or "I am held to account by the religion of love," because the d-y-n root yields both religion and judgment (compare yawm al-din, day of judgment). Targum's apparatus preserves both readings.

Frame-relation footnotes. The engine's footnote names the aspect-shading-into-priority relation between zahir and batin: the catalog of receptacles (pasture, convent, temple, Ka'ba, tablets, codex) enacts the qalb-as-mirror doctrine in which the perfected heart is the locus where every form of divine self-disclosure lands.

Glossary self-knowledge. The engine emits five `glossary_updates_proposed` entries: qabil, sura, qalb, din, and hubb. These are the terms a glossary expansion should target.

Verification layer results: schema validation passed; drift audit returned five lexicon-inconsistency incidents arising from surface-form mismatches between the engine's emitted `range_cards` sense IDs (English glosses) and the glossary's canonical kebab-case IDs; registry check returned

clean; citation check returned all four citations as training-memory (Nicholson 1911, Chittick 1989, Sells 1994, Chodkiewicz 1993), reflecting honest editorial discipline that page numbers from training memory are not the same as page numbers verified against printed editions.

6.3 Test B — kuntu kanzan hadith

The engine's primary rendering:

> I was a hidden treasure, so I loved to be known, and so I created creation in order to be known.

The substantive value-add over a generic LLM translation is concentrated in the apparatus.

The vocalization ambiguity. The Arabic verb a'rifa / u'rafa is written without diacritics (أعرف) and is morphologically licensed for either active ("that I [come to] know [Myself]") or passive ("that I be known"). The Akbarian commentary tradition explicitly preserves both. The translation leans passive (the standard reception); the active reading is preserved as an ambiguity entry with frame notes.

The attribution problem. Targum's apparatus carries a flagged footnote that the kuntu kanzan tradition is contested as hadith: not in the canonical Sunni or Shia collections; classified by al-Sakhawi (al-Maqasid al-Hasana #837) and al-Ajluni (Kashf al-Khafa II.132 #2016) as having "no transmissional origin"; declared not from the Prophet by Ibn Taymiyya (Majmu' al-Fatawa XVIII.376); classified as fabricated by al-Albani (Silsilat al-Ahadith al-Da'ifa #4101). The doctrinal weight in the Akbarian tradition is independent of the hadith-critical status: Ibn Arabi treats it as kashfi transmission, received via mystical unveiling rather than isnad. The Hekhal editorial position, enforced by the apparatus contract: surface the attribution problem on every appearance; never present the saying as canonical hadith; preserve its actual doctrinal function as Akbarian-tradition koan. A generic LLM translation, asked to produce English, will not surface this problem unprompted; the apparatus contract is what forces it.

Verification layer results: schema validation passed after one cycle (an enum violation in `ambiguities[].type` was caught and corrected); drift audit returned three lexicon-inconsistency incidents (same class as Test A); registry check returned two incidents — both `cross-reference-not-found` for Quranic verse refs (`texts/quran-5-54` and `texts/quran-51-56`) that the engine emitted as plausible Hekhal page slugs but that no Hekhal page implements. This is an honest engine failure mode: the engine extrapolated a Hekhal cross-reference convention that has not yet been authored. The fix is straightforward: either author the Hekhal pages, or surface the references via `glossary_updates_proposed` instead of `cross_references`. Citation check returned all seven citations as training-memory.

6.4 Test C pre-scaffolding (run-20260508T051332Z)

The original adversarial Test C was performed before the kataphatic-apophatic frame controller and the Christian-apophatic glossary existed. The engine produced a primary translation that, by itself, is competent:

> Trinity beyond being and beyond divinity and beyond goodness, overseer of Christian theosophy, direct us toward the unknown-beyond-unknowing and brilliant-beyond-light and uttermost summit of the mystical oracles.

This is a defensible apophatic-foregrounded rendering. The triple ὑπερ- (ὑπερούσιε / ὑπέρθεε / ὑπεράγαθε) is correctly identified as the operation that must survive in English. The apparatus material is substantive and well-attested: footnotes cite Andrew Louth's Denys the Areopagite (1989), Denys Turner's The Darkness of God (1995), and Paul Rorem's Pseudo-Dionysius commentary (1993).

Verification layer results at the time of the original run: schema validation passed; drift audit was clean (vacuously — the glossary was empty, so there was nothing to drift against); registry check and citation check did not yet exist. Verification layer results when re-run today against the current registry-check and citation-check implementations:

- Registry check: three incidents. (1) `range_cards[5].lexicon_ref: 'lexicon/koryphe'` — no Hekhal lexicon page implements this slug, and the term koryphe is not in any glossary file; this is a genuine fabrication. (2) `apparatus.cross_references[5].target_ref: 'lexicon/koryphe'` — same fabricated reference. (3) `apparatus.cross_references[6].target_ref: 'dionysius.mystical-theology.1.3'` — malformed reference: not in the `lexicon/{slug}` or `texts/{slug}` shape required by the cross-reference contract; the model attempted a chunk-id-style reference where a Hekhal-page-slug reference was expected.
- Citation check: three citations (Louth 1989, Turner 1995, Rorem 1993) all flagged as training-memory.

A subsidiary finding that emerged during the registry-check development pass is worth surfacing explicitly. The original analysis of this run (drafted into `benchmarks/findings-draft.md`) claimed six fabricated `lexicon_ref` references. When measured against the current Hekhal lexicon, only one of the six is genuinely fabricated: the other five (`lexicon/hyperousios`, `lexicon/theosophia`, `lexicon/logia`, `lexicon/hyperagnostos`, `lexicon/hyperphaes`) now resolve correctly because Hekhal lexicon pages have been authored in the interval between the original run and this paper's drafting. The honest reading is therefore narrower than the earlier framing: the engine emitted references the model expected the project to have, and the project caught up with the model's expectations on five of six. The one remaining fabrication, koryphe, is a real registry-check incident and the kind the new verification layer is built to catch.

6.5 Test C post-scaffolding (run-20260516T084703Z)

The kataphatic-apophatic frame controller (289 lines of YAML; six named relations: sequence, coincidence, excess, silence, tension; refusal clauses against importing non-Christian apophatic vocabulary into Christian-exegetical bodies) was shipped 2026-05-08. The Christian-apophatic glossary (15 entries: agnosia, apophasis, ekstasis, gnophos, henosis, hyperagnostos, hyperousios, hyperphaes, kataphasis, logia, mystikos, nous, theoria, theosis, theosophia) was shipped in the same window. We re-ran Test C with both firing.

The engine's post-scaffolding primary rendering:

> Trinity beyond being and beyond divinity and beyond goodness, overseer of the Christians' divine wisdom, direct us toward the unknown-beyond-unknowing and brilliant-beyond-light and uttermost summit of the hidden oracles.

The qualitative substantive change driven by the glossary's forced rendering for mystikos: the pre-scaffolding "mystical oracles" becomes the post-scaffolding "hidden oracles," reflecting the

glossary's enforced choice to honor the pre-12th-century mystery-cult etymology of mystikos over the later affective sense.

`frame_controllers_applied` is now `["kataphatic-apophatic", "quadriga"]`; both controllers exist on disk and validate clean. The six `range_cards` reference `lexicon_ref` fields that resolve against the current Hekhal lexicon. The `glossary_updates_proposed` array now proposes additions for hypertheos, hyperagathos, and ephoros with attested rationale and source citations — the model identifies coverage gaps rather than papering over them.

Verification layer results: schema validation passed; drift audit returned two lexicon-inconsistency incidents arising from surface-form divergence between the model's English-gloss `selected_sense` strings ("beyond being") and the glossary's canonical kebab-case sense IDs ("beyond-being"); registry check returned clean; citation check returned all five citations as training-memory.

Layer 4.5 cross-translator memory: we authored a verified PD index entry for Rolt 1920 Mystical Theology I.1 against the current Hekhal benchmark chunk-id convention, then re-ran the deterministic scorer over the post-scaffolding output. The scorer returned one entry (`translator: rolt, year: 1920, agreement: divergent, jaccard: 0.14`) with the verified Rolt rendering ("Trinity, which exceedeth all Being, Deity, and Goodness! Thou that instructeth Christians in Thy heavenly wisdom! Guide us to that topmost height of mystic lore which exceedeth light and more than exceedeth knowledge…") and the divergence note attesting that the Targum rendering uses the modern apophatic "beyond" while Rolt uses the Edwardian "exceedeth," producing low token-Jaccard overlap despite substantive doctrinal agreement. This is the expected behavior: token-Jaccard captures surface similarity, not doctrinal alignment; the scorer's value is in flagging which PD renderings the engine substantially overlaps with versus diverges from, not in claiming any single overlap measure carries doctrinal weight.

6.6 Independent adversarial review (Test C post-scaffolding)

A cold-context reviewer (Anthropic Claude in a fresh agent session, with no knowledge of the experimental hypothesis, the project, or the desired finding) was given the prompt and the output and asked to judge prompt-to-output compliance across eight dimensions. The reviewer's report (full text preserved in `benchmarks/dionysius-mt-1-1-opening/run-20260516T084703Z/adversarial-review.md`) found:

- **Glossary compliance:** mostly-compliant. All six glossary terms in the source receive selected renderings; no forbidden renderings appear. Out-of-glossary terms (hypertheos, hyperagathos) are surfaced honestly via `glossary_updates_proposed`, "exactly the prompt-specified behavior for out-of-glossary terms."
- **Frame-controller compliance:** compliant. Both controllers named with their enumerated relations (excess and dilation). Hyper- compounds preserved morphologically per the rendering rules. No refusal clauses violated.
- **Schema-field completeness:** mostly-compliant. The reviewer flagged that the output emits a populated `source.morph_stream` when the prompt instructs emitting `[]`; this is in fact the orchestrator's post-validation hygiene step that overwrites whatever the model emitted with the Layer 1 adapter's actual output. The reviewer correctly identified that the field is populated; the explanation (it is server-stamped, not model-emitted) is a layer below what the cold-context reviewer could see from the artifacts alone. This is genuinely informative about the engine's documentation surface: a reviewer reading prompt and output

without orchestrator source code will read the populated `morph_stream` as model non-compliance, when it is in fact orchestrator design.

- **Ambiguity handling:** compliant. The hyper-prefix-scope ambiguity (apophatic excess vs. Latin scholastic eminence) is surfaced with both readings, frame-relation tags, and substantive editorial note.
- **Citation grounding:** mostly-compliant. The reviewer verified all five cited works (Louth, Turner, Rorem, Bonaventure, Lubac) as real, canonical works in the field with plausible page ranges. No fabrications detected. Note that this is plausibility verification by independent reviewer, not the page-by-page verification the citation check requires for `verified` status; the citation check correctly reports all five as still training-memory pending editor sign-off against printed editions.
- **Apparatus quality:** compliant. Footnotes name controller and relation, explain the Greek's operation, connect to downstream textual events. Decision notes document choices with reasoning.
- **Translation fluency vs fidelity:** compliant. The reviewer judged the primary translation defensible as both English prose and a translation of the Greek source.
- **Overall:** "Strong, editor-grade output… suitable for editor sign-off with two trivial fixes" (the morph_stream presentation issue and a borderline-expansive set of cross-tradition cross-references to Akbarian and Kabbalistic counterparts).

The cold-context review constitutes evidence that the post-scaffolding output meets the substantive compliance bar set by the prompt's contract, evaluated without contamination from the experimental hypothesis or the desired finding. We do not claim more than this from a single review: a production-grade evaluation requires multiple independent reviews across multiple passages.

6.7 Results matrix

Run	Schema	Drift	Registry	Citations (V / TM / U)	PD-comp
Test A re-run (post-fix)	pass	5 incidents	0	0 / 4 / 0	(chunk not in PD index)
Test B re-run (post-fix)	pass (1 retry)	3 incidents	2 incidents	0 / 7 / 0	(no PD English for hadith)
Test C pre-scaffolding (original)	pass	clean (vacuous)	3 incidents	0 / 3 / 0	(no PD index coverage)
Test C post-scaffolding (this report)	pass	2 incidents	0	0 / 5 / 0	1 entry (Rolt, divergent, J=0.14)

V = verified against printed/scanned source; TM = training-memory (real work, page unverified); U = unverified (not in manifest).

7. The article finding: scaffolding plus registry-grounding matter more than the schema

The Phase-1 architectural commitments distill into a two-part empirical claim.

First, **the engine's distinctive value over generic LLM translation is realized when the scaffolding for the source's domain is built**. When the scaffolding fires (Tests A, B, and the post-scaffolding Test C), the engine produces output whose distinctive value over standard PD translations is real and locatable: ambiguities surfaced, frame relations named, glossary updates proposed, attribution problems flagged, cross-tradition cross-references emitted, drift audit catching real surface-form divergence. Independent adversarial review judges the post-scaffolding output editor-grade. The scaffolding does not replace the translator; it concentrates the translator's discipline into the structured fields, where it becomes interrogable.

Second, **schema-shape enforcement is necessary but not sufficient; registry-grounding is the second leg**. JSON Schema enforces shape and field types; it cannot enforce that a `lexicon_ref` points to an extant entry, that a `frame_controllers_applied` entry corresponds to an on-disk controller, or that an `audit_trail.lexicon_revision` names a glossary revision that has been published. The pre-scaffolding Test C demonstrates the failure mode the schema alone cannot catch: a schema-valid output references infrastructure that does not exist. The Phase-1 registry-check pass catches one genuinely fabricated lexicon reference (koryphe) and one malformed cross-reference in this output, while correctly not flagging five other references that have been authored in the interval — the project caught up with the model's expectations on five of six, and the registry check is the mechanism that distinguishes the two cases. The same registry-check pass catches two fabricated Quranic cross-references in the post-scaffolding Test B output, demonstrating that even fully-scaffolded runs benefit from registry-grounded verification.

The practical implication for the coverage strategy is twofold: **the scaffolding (glossary, frame controllers, scholarly-corpus indexing) must be built for every Tier-1 corpus before that corpus ships translations through Targum**; and **the registry-check and citation-check passes must be wired into the editor sign-off workflow as required gates for the `verified` status flip**, not optional advisories.

---

8. Limitations honestly named

We name five limitations bounding the Phase-1 claims.

8.1 Single-run benchmarks. Each of the three flagship benchmarks is one generation call. Production-quality evaluation requires multiple runs per passage to characterize variance, plus held-out scoring against a larger evaluation set. The mystical-register evaluation set is scoped at approximately 500 passages across the Tier-1 corpora; we have not formally built it. The Phase-2 commitment is the evaluation harness plus a public agreement dashboard.

8.2 PD-comparison chunk-id mismatch (partly resolved). The PD-comparison index originally had a chunk-id convention that did not align with the benchmark spec convention used by the three flagship tests; the deterministic scorer had zero coverage of Tests A, B, C. As part of this paper's preparation we authored a verified PD index entry for Rolt 1920 MT I.1 against the benchmark-spec convention, demonstrating the fix. The remaining work is normalization: an automated chunk-id alignment step at index load time so the PD index covers all benchmark passages

without per-entry authoring. The kuntu kanzan hadith remains uncovered because no PD English exists for it; this is a corpus gap the AI-assisted translation pipeline is designed to close as part of the coverage strategy.

8.3 Vacuous-compliance detection works but depends on infrastructure currency. The registry-check pass catches infrastructure references that do not resolve. A subsidiary finding documented in §6.4 is that infrastructure references the model emitted in 2026-05-08 are partially "caught up to" by 2026-05-16 as the project's lexicon pages are authored. This is a feature, not a bug: it means registry-grounded verification rewards the project for shipping infrastructure rather than penalizing the engine for anticipating it. But it also means the registry-check incident counts reported in §6 are time-stamped: the same outputs measured against the project's infrastructure at different points in time will return different incident counts.

8.4 PD-quote verification. Reference translation quotations reproduced from training memory must be confirmed against archive.org or CCEL scans before publication. We verified the Rolt 1920 Mystical Theology I.1 prayer against the CCEL hosted edition; the Parker 1897 Mystical Theology I.1 prayer and the Nicholson 1911 Tarjuman al-Ashwaq XI.13–15 are not verified against the canonical scans (Parker Vol. II not located in the OCR scans I had access to; Nicholson archive.org OCR mangled past grep). The Hekhal editorial discipline, documented in `benchmarks/pd-quote-verification.md`, requires every PD quote in publication-ready output to trace to one of: a verified scan; a clearly-attributed secondary source that quotes the original; or a regenerated `comparison_to_pd_translations[]` entry derived from text fetched verbatim from a verified scan via the deterministic scorer.

8.5 In-session-LLM generation pathway. The Test A, B, and C post-scaffolding re-runs reported here were generated via an in-session-LLM workflow (`scripts/manual_run.py`) rather than via the API path. This workflow is the project's natural fit for the bulk-translation coverage strategy: it uses the controlled-vocabulary, frame-controller-aware, schema-validated pipeline without per-call API spend. We are explicit that it is not a controlled experimental condition for the comparison reported in §6.4–6.5: the in-session generation pathway differs from the API path in three material ways (the model sees the prompt as conversational context rather than as a tool-use directive with `input_schema` enforcement; the generating session may have contextual knowledge of the project that the API path does not; statistical sampling via temperature variance is not available). The verification layer evidence (registry check, citation check, independent adversarial review) provides post-hoc grounding that the in-session output meets the compliance bar the prompt sets, but does not substitute for the controlled experimental conditions that multi-seed API runs would provide. The Phase-2 commitment is to industrialize this workflow with a `SessionLLMClient` adapter and to run side-by-side API/in-session comparisons once the API budget is restored.

---

9. Reproducibility, openness, and methodology notes

9.1 Artifact preservation and openness scope

Every benchmark run preserves a full audit package at `benchmarks/{spec-id}/run-{timestamp}/`: the spec snapshot, the assembled prompt (system + user), the validated output JSON, the drift report, the registry report, the citation report, and (for Test C post-scaffolding) the adversarial review. The artifacts that are necessary to inspect-verify the empirical

claims of this paper — the three benchmark specs, the original (2026-05-08) and re-run (2026-05-16) outputs, the verification reports, the adversarial-review transcript, a verified-citation manifest excerpt, and the pre/post-scaffolding comparison — are published as a public reproducibility surface at hekhal.org/targum-experiments/scaffolding-matters-paper. A reader can audit the engine's compliance with the discipline contract surfaced in §6 without access to the engine's source code; the audit packages are the evidence.

The Targum engine source code, the controlled glossaries, the frame controllers, and the curated scholarly-corpus seeds are not part of the public reproducibility surface. They are the proprietary editorial work of the Hekhal Project, which is a commercial entity (Lattice DBA, EIN issued 2026-04-02) and they constitute, in aggregate, an editorial moat that the project's coverage strategy depends on for long-term sustainability. The decision to keep these surfaces closed is a deliberate one; the open-source presumption that runs through digital-humanities methodology work does not, in our view, apply uniformly to engineering work whose competitive value lies in the per-corpus editorial scaffolding rather than in the algorithmic novelty. The engine source code, the glossaries, the frame controllers, and the scholarly-corpus seeds are available to the journal's peer-review process under standard reviewer-disclosure terms, on request to the corresponding author. Future versions may release subsets of this material under restrictive licenses (academic-use-only; named-collaborator-only) once a sustainable funding model is in place.

The architecture (§3), the seven Phase-1 commitments (§4), the registry-grounded translation discipline (§7), and the verification-layer methodology (§6.1) are described at the level required to replicate the design in an independent implementation. A research group choosing to build a controlled-vocabulary translation engine for a different esoteric corpus from the descriptions in this paper would not, in our judgment, require the Hekhal engine's source code to do so. The contribution this paper makes to the field is the design and the empirical findings, not the implementation.

9.2 Three paths for generation

Phase 1 ships two paths for the generation step; the original Phase-2 commitment scoped a third path that has shipped subsequent to this paper's first drafting and is included here for completeness.

(a) **API client.** The orchestrator calls `anthropic.Anthropic().messages.create()` with the assembled prompt as the user message and the schema as a tool `input_schema`. This is the production path for paper-grade controlled experiments where multi-seed variance characterization and model-version-drift detection are required; it consumes API credits.

(b) **In-session-LLM manual workflow.** `scripts/manual_run.py` splits `benchmark.run()` into a `prepare` phase (assemble prompt, write run dir, print prompt to stdout) and a `finalize` phase (validate the LLM's JSON response, run all post-processing, write outputs including the registry and citation reports). The generation step happens manually: a human pastes the assembled prompt to an LLM, the LLM emits the TranslationOutput JSON, the human pipes the response back to the script. The result is structurally indistinguishable from an API-driven run, differing only in `audit_trail.model` (stamped `"claude-opus-4-7[session]"` rather than the API equivalent). This is the path used for the three re-runs reported in this paper.

(c) **In-session-LLM file-pipe workflow (SessionLLMClient).** Shipped 2026-05-16 as the first item in the Phase-2 backlog. The `SessionLLMClient` is a proper `LLMClient`

implementation registered with the orchestrator under `--client session`; it writes the assembled prompt envelope to a watched directory (`pipe/inbox/{request-id}.prompt.json`) and polls a sibling directory (`pipe/outbox/{request-id}.response.json`) for the operator's response. The orchestrator drives `benchmark.run()` (or `translate`, `synthesize`, `eval`) end-to-end without the prepare/finalize copy-paste hop, while the in-session human plays operator on the file-pipe. The post-validation finalize logic (morph clobber, audit-trail stamps, PD-comparison rebuild, gematria detection, drift / registry / citation checks, comparison-scaffold render) was lifted into a shared helper so paths (a) and (c) produce structurally identical run-dir artifacts; the API path additionally now runs the full Phase-1 verification stack, which it previously did not.

We used path (b) for the experimental re-runs in this paper because path (c) shipped after the experimental design was frozen. Path (c) is the production path for the bulk-translation coverage strategy: driving genuinely first-ever PD-license English translations of esoteric primary texts that have no PD English at all — for example the kuntu kanzan hadith itself, the Akbarian secondary corpus, the Hesychast literature past Philokalia — through the same controlled-vocabulary, frame-controller-aware, schema-validated, registry-grounded, citation-verified pipeline that produced the benchmark results. The economics of path (c) shift the cost of bulk-translation work from per-call API spend to existing subscription tokens, which makes the long-tail coverage strategy economically tractable as a sustained personal practice. Future versions of this work will report controlled side-by-side comparisons of paths (a) and (c) to characterize whether the two paths produce materially different output under the same scaffolding conditions; we have not done that comparison in this paper and we are explicit that path (b)'s use here is not a substitute for the controlled-experimental conditions a multi-seed (a) versus (c) study would provide.

9.3 Engineering changes made during Phase 1 and early Phase 2

Seven engineering improvements landed during the Phase-1 evaluation window and the first Phase-2 pass that immediately followed.

`backend/benchmark.py` now stamps `audit_trail.drafted_at` from the actual run time and computes `audit_trail.prompt_hash` from the assembled prompt. A shared `finalize_run()` helper (Phase 2, P2.1 ship) consolidates morph clobber, audit-trail stamping, PD-comparison rebuild, gematria detection, drift / registry / citation checks, and comparison-scaffold rendering, so the API and in-session-file-pipe paths produce structurally identical run-dir artifacts; the API path now also runs the full Phase-1 verification stack, which previously stopped at the drift audit.

`backend/orchestrator.py` and `backend/prompt.py` now jointly handle the `morph_stream` fabrication problem: the schema is tightened to `MorphToken.additionalProperties: false`; the prompt explicitly instructs the model to emit `source.morph_stream: []`; and the orchestrator pre-validation hygiene step clobbers any stray morph tokens with the deterministic adapter output. The post-validation hygiene was discussed in §6.6: a cold-context reviewer reading prompt and output without orchestrator source code will see the populated morph_stream as model non-compliance, when it is in fact orchestrator-level design.

`backend/apiclient.AnthropicClient.generate()` catches `anthropic.InternalServerError`, `RateLimitError`, and `APIConnectionError` and re-raises them as `APIClientError` so the benchmark retry loop can handle them with exponential backoff.

`backend/apiclient.SessionLLMClient` (new, Phase 2 P2.1) implements the file-pipe LLM client described in §9.2 path (c).

`backend/registry_check.py` (new) implements Layer 6.5: post-LLM verification that every infrastructure reference in a TranslationOutput resolves to extant infrastructure (frame controllers on disk, lexicon pages on the Hekhal site, glossary terms, glossary revisions).

`backend/citation_check.py` (new) implements Layer 6.6: every output citation is checked against `corpus/verified_citations.yaml` and labelled `verified`, `training-memory`, or `unverified`.

`backend/lexicon_sync.publish_stubs()` (new, Phase 2 P2.6) closes the registry-check coverage gap from the editor's side: every glossary term gains a Hekhal-shaped lexicon MDX stub with canonical front-matter (term, transliteration, script, language, tradition, glossEnglish, relatedTerms, seeTexts) and a sectioned body the editor expands. The architectural point is small but operationally important: registry-grounded verification is only useful to the extent the registry it grounds against is well-populated, and stub generation amortizes the cost of getting there.

---

10. Open questions

We close with five questions Phase 2 will engage.

Model drift over time. A controlled-vocabulary engine assumes the underlying LLM's behavior is stable enough that the same prompt produces materially similar output across model versions. Anthropic, OpenAI, and the open-model ecosystem release new model versions monthly. The audit trail's `audit_trail.model` field is sufficient to detect drift retrospectively; the evaluation harness must produce a metric for it.

Glossary maintenance at scale. The current glossaries (49 entries across 4 files) are small enough that one editor can maintain them by hand. At the projected scale (200+ entries across 8 Tier-1 corpora), maintenance becomes a workflow problem: when does a `glossary_updates_proposed` entry become a glossary entry, who approves it, how is the upstream change communicated to in-progress translations, how is glossary versioning surfaced in published Hekhal pages.

Editor sign-off at scale. Every Targum output ships as `status: machine-assisted` and requires human sign-off to flip to `verified`. The calibration is currently one editor (the author); at corpus-scale this becomes a multi-editor workflow with conflict resolution, citation discipline, and reviewer-track audit. The seven-state `TranslationStatus` taxonomy is in place; the workflow software is not.

Citation verification at scale. The verified-citation manifest discipline is sound but its operating cost scales linearly with corpus size. Phase 2 will scope automation candidates (Google Scholar / WorldCat lookup for verification candidates; OCR-grade matching against archive.org-hosted PD editions; secondary-source citation chains where an editor verifies the secondary source and the primary citation through it).

The mystical-register evaluation set. Building an evaluation set of approximately 500 passages across Tier-1 corpora where modern critical translations agree, then measuring engine-vs-critical agreement at token / phrase / structural / semantic levels, is the Phase-2 deliverable that

converts these case-study findings into statistical claims. The agreement dashboard summarizing per-corpus and per-translator score distributions will be released publicly as a credentialing surface; the underlying source-translation alignment dataset is part of the project's proprietary scholarly corpus and will be available to peer-reviewers and named-collaborator scholars under reviewer-disclosure terms, on the same footing as the engine source code.

Acknowledgments

The author thanks the open-source NLP communities responsible for CAMeL Tools (Arabic morphology), Dicta (Hebrew and Aramaic morphology), CLTK (Classical Languages Toolkit), and Stanza, which together make the Layer-1 morphology router possible across the language coverage Targum requires. The author thanks the Sefaria, Perseus, OpenITI, GRETIL, and CCEL projects, whose open primary-source corpora and reference editions are the foundation of the Layer-0 reference-resolution adapter system. The author thanks Anthropic for access to the Claude Opus 4.7 model, against which the engine is currently calibrated. The independent adversarial review reported in §6.6 was conducted by a fresh Anthropic Claude session in agent mode with no contextual knowledge of the experimental hypothesis or the project; this is an artifact of the Phase-1 methodology and the author is the sole human author of the present paper.

The author thanks the public-domain translation tradition (Nicholson, Mead, Parker, Rolt, Westcott, Mathers, Underhill, MacKenna, and many others), whose unrepublishable scope and reverent care define both the floor and the ceiling that the AI-assisted coverage strategy must meet. The author thanks the cross-tradition scholarly tradition (Chittick, Chodkiewicz, Sells, Idel, Scholem, Corbin, Louth, Turner, Rorem, Schimmel, and others) whose published English-language scholarship makes serious cross-tradition esoteric study a viable register of work, and whose scholarship the Hekhal project's Layer 3 retrieval corpus drinks deeply from.

The Hekhal project is supported through reader contributions and the project's affiliate revenue layer; no external grant or institutional funding has supported this work.

Data availability

Audit-package artifacts sufficient to inspect-verify the empirical claims of this paper are published at hekhal.org/targum-experiments/scaffolding-matters-paper. These comprise: the three benchmark specifications (Tests A, B, C), the assembled prompts, the validated TranslationOutput JSON for the original (2026-05-08) and re-run (2026-05-16) runs, the drift / registry / citation reports for each run, the adversarial-review transcript for Test C post-scaffolding, the verified-citation manifest excerpt, and the PD-quote verification log.

The Targum engine source code, the controlled glossaries, the frame controllers, and the curated scholarly-corpus seeds are not part of the public artifact set. They are available to the journal's peer-review process under standard reviewer-disclosure terms by request to the corresponding author (vinnycouey@gmail.com). The disclosure rationale is articulated in §9.1.

Competing interests

The author is the sole proprietor of the Hekhal Project (operated under the Lattice DBA umbrella, EIN issued 2026-04-02), which is the commercial entity that owns the Targum engine source code and the corpus-specific scaffolding described in this paper. The Hekhal Project generates revenue through reader contributions, an affiliate-bookshop layer, and (in scope for Phase 2) consulting work in which the Targum engine is applied to client corpora. The author's livelihood depends in part on the continued viability of these revenue streams. The author has no other financial or institutional interest that materially shapes the conduct or the reporting of this work; in particular, the author has no employment, consulting, or equity relationship with Anthropic, OpenAI, or any other model provider whose product is referenced in this paper.

Author biography

Vincent W. Couey is an independent researcher working at the intersection of digital humanities, esoteric primary-source translation, and small-scale computational philology. He founded the Hekhal Project in 2026 to produce a serious, open, cross-tradition reference for the mystical and contemplative primary-source corpus; the Targum translation engine, described in this paper, is the project's translation arm. His other current research includes Substrate Geometry (a computational physics program on rigid-body equilibria and shape-classification dynamics) and a computational toxicology program testing architecture-specific QSAR failures on psychedelic-class compounds (first preprint on ChemRxiv, OSF DOI 10.17605/OSF.IO/UWVX4). He works from Toledo, Ohio. Correspondence: vinnycouey@gmail.com; project: hekhal.org.

References

Chittick, William C. 1989. The Sufi Path of Knowledge: Ibn al-'Arabi's Metaphysics of Imagination. Albany: SUNY Press.

Chodkiewicz, Michel. 1993. An Ocean Without Shore: Ibn 'Arabi, the Book, and the Law. Translated by David Streight. Albany: SUNY Press.

Faivre, Antoine. 1994. Access to Western Esotericism. Albany: SUNY Press.

Gao, Yunfan, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2024. "Retrieval-Augmented Generation for Large Language Models: A Survey." arXiv:2312.10997v5.

Garcia, Xavier, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. "The Unreasonable Effectiveness of Few-Shot Learning for Machine Translation." Proceedings of the 40th International Conference on Machine Learning.

Hanegraaff, Wouter J. 2012. Esotericism and the Academy: Rejected Knowledge in Western Culture. Cambridge: Cambridge University Press.

Hendy, Amr, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. "How Good Are GPT Models at Machine Translation? A Comprehensive Evaluation." arXiv:2302.09210.

Kockaert, Hendrik J., and Frieda Steurs, eds. 2015. Handbook of Terminology, Volume 1. Amsterdam: John Benjamins.

Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Advances in Neural Information Processing Systems 33.

Louth, Andrew. 1989. Denys the Areopagite. London: Continuum.

Nicholson, Reynold A. 1911. The Tarjumán al-Ashwáq: A Collection of Mystical Odes by Muhyiddín Ibn al-'Arabí. London: Royal Asiatic Society.

Parker, John. 1897. The Works of Dionysius the Areopagite. London: James Parker.

Rolt, C. E. 1920. Dionysius the Areopagite: On the Divine Names and the Mystical Theology. London: SPCK.

Rorem, Paul. 1993. Pseudo-Dionysius: A Commentary on the Texts and an Introduction to Their Influence. Oxford: Oxford University Press.

Schäffner, Christina, ed. 2002. The Role of Discourse Analysis for Translation and in Translator Training. Clevedon: Multilingual Matters.

Sells, Michael. 1994. Mystical Languages of Unsaying. Chicago: University of Chicago Press.

Turner, Denys. 1995. The Darkness of God: Negativity in Christian Mysticism. Cambridge: Cambridge University Press.

Versluis, Arthur. 2007. Magic and Mysticism: An Introduction to Western Esotericism. Lanham, MD: Rowman & Littlefield.

Vilar, David, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. "Prompting PaLM for Translation: Assessing Strategies and Performance." Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics.

Wang, Liang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. "Multilingual E5 Text Embeddings: A Technical Report." arXiv:2402.05672.

Public reproducibility package: hekhal.org/targum-experiments/scaffolding-matters-paper. Engine source disclosure on request to corresponding author. Editorial correspondence: vinny-couey@gmail.com.