

UTS at CheckThat! 2026: Cite-Frame Engineering for Generated Fact-Checking Articles

Dima Galat¹, Marian-Andrei Rizoiu¹

¹University of Technology Sydney, Australia

Abstract

CheckThat! 2026 Task 3 asks systems to generate fact-checking articles, graded by an unweighted mean of four sub-metrics (M4). **Our UTS submission placed 2nd of 11 teams** (M4 = 0.484). The shipped system is a deterministic stub drafter wrapped by two single-lever interventions: a domain-attribution cite frame (*HostCite*) and a shadow-validated anchor picker (*ShadowVal*) that use the Llama-3.2:1B large language model (LLM) only as a per-cite validator, never as a body-prose generator. The stack lifts M4 by +0.027 over the stub on the WatClaimCheck validation split and follows two design rules our ablation matrix made unambiguous. *Scorer conservatism*: credit only tokens the references entail — templates pay; LLM prose, reviewer names, and raw evidence all fail. *Auxiliary anchor signals are miscalibrated against the Llama judge*: every anchor proxy we tried (cross-encoder, length, lead position) picks anchors the judge rejects — gate on the judge itself. Our citation precision/recall (P/R) deficit (0.299 vs 0.671) outweighs our leading coverage and second-place entailment, leaving a +0.062 M4 gap to the winner — consistent with a *selective-emission* policy that drops low-confidence cites.

Keywords

CheckThat!, fact-checking article generation, natural language inference, citation evaluation, shared task

1. Introduction

CheckThat! 2026 Task 3 [1, 2] asks systems to generate a fact-checking article given a textual claim, its claimant, the original verdict, and a dictionary of pre-retrieved evidence URLs paired with their full extracted text; the article is graded as the unweighted mean of four sub-metrics (the leaderboard score, M4); three are computed against the article-as-emitted and one (entailment) against a held-out human reference. Because two of the three article-side metrics use a strict regular expression for parsing (source: URL) citations, and because the entailment metric chunks both the generation and the reference into fixed three-sentence units before running bidirectional natural language inference (NLI) with a RoBERTa model fine-tuned on the Multi-Genre NLI corpus (RoBERTa-MNLI) [3, 4], the *citation sentence* — not the article as a whole — is the unit of optimisation. Every design decision in this paper follows from that scorer mechanic.

The obvious baseline is to feed claim and evidence to an instruction-tuned large language model (LLM) and let it draft reviewer-style prose; we attempted this at three abstraction levels (full-body drafting with Qwen3:8B [5] on $n = 29$ items, full-body drafting with Qwen3:30B-A3B [5] on $n = 50$, and lede-only rewriting with Qwen3:30B-A3B on $n = 50$), and all three lose to a deterministic templated stub on M4. The loss is structural: LLM-generated chunks reliably miss the specific token patterns the reference distribution rewards under MNLI, and LLM cite-formatting drift triggers the citation parser strictly enough that single-item failures collapse the whole submission's citation precision and recall (P/R). Capacity does not fix this. Replacing the entire article with LLM output costs -0.026 M4 at 8B and -0.070 at 30B; preserving the body but rewriting only the lede still costs -0.011 . Prompt engineering alone does not break this; closing the gap needs a different mechanism (e.g., constrained decoding to force verbatim cite anchors [6]).

Two narrow, deterministic interventions on top of the stub drafter account for the bulk of our headline lift. **HostCite** wraps each citation in a domain-attribution frame (According to {host}, {anchor})

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

✉ dima.galat@student.uts.edu.au (D. Galat); Marian-Andrei.Rizoiu@uts.edu.au (M. Rizoiu)

🆔 0000-0003-3825-2142 (D. Galat); 0000-0003-0381-669X (M. Rizoiu)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

(source: {url}) .), preserving the article’s chunk count while pushing each cite chunk closer to the reviewer-style distribution that drives MNLI entailment. **ShadowVal** runs the same Llama-3.2:1B [7] judge that the scorer uses to validate the stub’s anchor choice; if Llama says NO, we escalate to the next non-boilerplate sentence in the evidence, and if all candidates fail, we fall back to the stub anchor unchanged, which by design means ShadowVal never lowers the per-cite judge’s acceptance rate relative to the underlying anchor picker. The two stack sub-additively: both improve P/R on cite-sentences but on overlapping items, so ShadowVal adds only +0.0075 on top of HostCite’s +0.0199 standalone (versus +0.0145 standalone for ShadowVal). The stack still clears $K = 1$ by +0.0274 M4. Both interventions respect the single principle that explains every kill in our ablation matrix: *the scorer’s two sub-judges only credit tokens they can verify against their references*, expressed concretely as chunk-count preservation on the MNLI side and cite-sentence entailability on the Llama side (§3.4).

This paper reports one empirical finding and two design rules on a system that placed 2nd of 11 teams. The *empirical finding* is that our citation P/R deficit (0.299 vs 0.671) outweighs our entailment (2nd) and coverage (1st) advantages, leaving a +0.062 M4 gap to the winner; the candidate mechanism is *selective emission* — declining to emit cites the per-cite judge would refuse — which we did not test during the campaign and flag as the highest-priority refutation for the next cycle (§6.1). The two *design rules* drove the system’s contributions. **Design rule 1: scorer conservatism.** Both sub-judges (MNLI on chunks, Llama on cites) refuse to credit tokens they cannot verify against their references. Respecting this rules out a large class of intuitively-attractive interventions — LLM-style lede rewriting, reviewer-name injection, padding chunks with raw evidence — and explains every kill in our ablation matrix. We name two concrete instances: chunk-count preservation on the MNLI side, cite-sentence entailability on the Llama side (§3.4). **Design rule 2: auxiliary anchor signals are miscalibrated against the judge.** Cross-encoder claim-relevance, length-preferring heuristics, and even lead-position selection all pick anchors that sound right to a human but the Llama judge refuses to entail standalone. Mirroring the judge during anchor selection — calling the same Llama with the same prompt and accepting only its YES verdicts — recovers the loss (§4.4). The accompanying artefacts are a deterministic pipeline that uses Llama-3.2:1B only as a shadow validator (never as a generator) and reaches test M4 = 0.484, and thirteen documented ablations with mechanism-level explanations grounded in the two design rules. §2 sketches the task and prior baselines, §3 dissects the scorer mechanics that determine our design space, §4 walks the system pipeline, §5 reports the experiment matrix including the official leaderboard, and §6 discusses the headline P/R gap, the selective-emission hypothesis, and what generalises beyond CheckThat!.

2. Task and Related Work

Task definition. Each input item provides a claim, claimant, original verdict label, and a dictionary mapping evidence URLs to their extracted page text; the test set contains 1158 items. The required output is a free-form article in which every cited statement is followed by a (source: URL) suffix. Scoring averages four sub-metrics: chunked bidirectional RoBERTa-MNLI entailment between the generation and a held-out reference (§3.1); citation precision and recall, judged per cite-sentence by Llama-3.2:1B (§3.2); and evidence coverage as the fraction of provided evidence URLs cited at least once (§3.3).

Validation data. Training and validation derive from WatClaimCheck [8], with a 3372-item validation split (henceforth *val*) that includes human reference articles. We work on a 200-item stratified subsample for fast iteration (described in §4.5); human references for the test split are not released to participants, which means the test-side entailment sub-metric is unobservable until the post-competition reveal and any leaderboard inference from the visible 3-metric sub-score must be calibrated against *val*.

Prior baselines. The lab provided one organiser baseline and one external reference submission, the IFM_MBZUAI K2-V2-Instruct sample [9]; we compare against the former on the official leaderboard (§5.5) and analyse the latter’s citation-format failures in §5.6.

Related work. Automated fact-checking has an established benchmark literature on claim verifica-

tion [10, 11], and the CheckThat! lab has run claim-verification subtasks for several editions [12]. Closer to Task 3 is the line of work on generating *justifications* for fact-checking verdicts: extractive–abstractive explanation generation over PolitiFact articles [13], explainable verdicts for public-health claims [14], and few-shot, retrieval-augmented justification generation on the ExClaim dataset [15], which also contributes part of the Task 3 test collection. Sahnan et al. [16] introduced the task of generating *full* fact-checking articles and proposed QRAFT, an LLM-based agentic framework that mimics the writing workflow of human fact-checkers; their human evaluations found LLM-drafted articles still lag expert-written ones. Task 3 in 2026 is the first shared task built around this article-generation formulation, and the metric design (chunked NLI plus LLM-judged per-cite) distinguishes it from prior generation-style benchmarks where reference-free LLM judges [17] or BERTScore-style overlap [18] drive scoring.

3. Task Scoring Mechanics

This section dissects the scorer because the scorer’s mechanics, not fact-checking in general, determine our design choices. Every lever in §4 and every kill in §5.4 traces back to a specific behaviour of the four sub-metrics described here.

3.1. Entailment

The entailment sub-metric runs bidirectional NLI (`roberta-large-mnli`) between the generation and the human reference, both segmented into three-sentence chunks. The A-side score is the mean over generation chunks of the max entailment probability to any reference chunk (precision-flavoured); the B-side is symmetric over reference chunks (recall-flavoured); the final metric is their average. The dominant design lever is reference-style mimicry: chunks that read like the reference distribution score higher regardless of factual content alignment, because surface token patterns drive the MNLI head.

3.2. Citation precision and recall

Each cited sentence is identified by a (`source: URL`) suffix and passed to Llama-3.2:1B [7], served by ollama [19]; the prompt asks YES/NO whether the cited sentence is supported by the evidence text. Recall is the rate of YES verdicts across all cite-sentences; precision is the conditional rate, equal to recall when every cite has a single URL. Two properties of this design are load-bearing: (a) single-URL cites avoid the precision-dilution risk of multi-URL bundles whose constituent URLs may not all be supported; and (b) Llama-3.2:1B at this scale is generous and surface-pattern-driven: short literal cites against long premises usually pass, but meta-claims about the fact-check article itself (reviewer name, verdict, date) get rejected. We pin the model digest for byte-deterministic scoring (Appendix A).

3.3. Evidence coverage

Coverage is a set-membership metric: the fraction of provided evidence URLs that appear at least once in the union of cited URLs across the article. It approaches 1.0 when a system emits at least one cite per provided URL, modulo parser-regex strictness on individual cites. Coverage is therefore not a discriminating signal between top-tier teams (§5.5), but it is the single largest gap between the lab baseline and any team, a precondition for top-tier placement rather than a differentiator.

3.4. Scorer conservatism

The ablation matrix (§5.4) yielded a single principle that explains every kill on the metric side: *both sub-judges (MNLI on chunks, Llama on cites) only credit tokens they can verify against their respective references; tokens they cannot verify drag the sub-metric down*. The principle has two concrete instances, which we name and reuse as design rules in §4.

3.4.1. Chunk-count invariance (MNLI side)

The entailment metric averages max-entailment-probabilities over generation chunks (A-side) and reference chunks (B-side). Adding a chunk whose content does not match any reference chunk lowers the A-side average. This is not a categorical “more chunks is bad” rule — a chunk that does entail to some reference chunk improves the B-side — but in practice templated reviewer-style prose we can generate does not match the actual reference distribution. *HostCite+Body* added one templated middle sentence to the lede and lost -0.032 entailment versus *HostCite*; the safe move on this metric is to preserve chunk count and enrich existing chunks in-place.

3.4.2. Cite-sentence entailability (Llama side)

The Llama judge runs over a (cite-sentence, evidence-text) pair. Any token in the cite that the source URL’s evidence does not support triggers NO. *HostCite+Reviewer* added the clause “in a review by {reviewer_name}” inside the cite and lost -0.037 P/R: the source URL does not entail meta-claims about who reviewed it, so Llama correctly refused. The cite frame may add only source-attribution tokens (According to, states that, In a report from) that are entailable from the cited URL itself; reviewer names, verdict labels, and dates injected into the cite sentence are uniformly rejected. We use this as a hard constraint when designing cite-frame variants in §4.3.

4. System

4.1. Overview

The system maps each input item (claim, claimant, rating, evidence URL→text dictionary) to a single fact-checking article string with embedded (source: url) citations. The core design choice is a deterministic stub drafter; we make one LLM call per evidence URL for shadow validation of the cited anchor sentence, and no LLM calls in the body prose.

4.2. Stub generator: deterministic K=1 baseline

The stub drafter emits, for each input item, a lede sentence, one cite sentence per provided evidence URL using the first non-boilerplate sentence of the evidence as the anchor, and a single conclusion sentence. The cite syntax is (source: url); when a cite carries multiple URLs, they are separated by semicolons, the only separator the scorer’s parser accepts. We refer to this configuration as $K = 1$, where K is the number of cite sentences emitted per provided evidence URL (a $K = 2$ variant emitting two anchors per URL is ablated in §5.4). The motivation is that any submission that fails the citation-format regex on even a single item zeroes that item’s P/R contribution, and a deterministic format-checker-correct floor is the precondition for any cite-frame lever that comes later. By construction the stub emits one cite per evidence URL, achieving near-maximal coverage in practice (0.987 visible, 0.996 official) modulo parser-regex strictness on a handful of cites; its precision and recall are determined entirely by anchor-picker quality and Llama’s judging variance.

4.3. HostCite: domain-attribution cite frame

Each citation is wrapped as a single sentence According to {host}, {anchor} (source: {url}) ., where {host} is the registered domain extracted from the URL and {anchor} is the anchor sentence chosen by the stub picker. The wrap is chunk-count-invariant (one sentence in, one sentence out) and adds only source-attribution tokens that mirror reviewer-style prose, without introducing the meta-claims (reviewer name, verdict, date) that the per-cite judge refuses (§3.4).

The chunked MNLI mechanic means each cite chunk’s entailment-to-reference correlates strongly with how reviewer-stylish that chunk reads; templated source-attribution prose is the cheapest de-

terministic way to nudge each cite chunk toward the reference distribution without violating either invariant.

At full val HostCite lifts M4 by +0.0199 over the stub, with a paired-bootstrap 95% confidence interval (CI) of [+0.0176, +0.0222]; the gain is dominated by precision/recall (+0.035) and a smaller but consistent entailment contribution (+0.010). HostCite is rating-mix-sensitive: its biggest gains come from the Mixture and Correct Attribution rating buckets (see §4.5).

4.4. ShadowVal: shadow-validated anchor selection

For each evidence URL the stub picker first proposes an anchor sentence; we then call Llama-3.2:1B with the verbatim scorer prompt on (proposed_anchor, evidence_text) and read the YES/NO verdict. If YES, we keep the proposed anchor. If NO, we escalate to the next non-boilerplate sentence in the evidence and re-call; if every candidate sentence fails, we fall back to the stub anchor unchanged. By design the fallback path ensures ShadowVal cannot lower the per-cite judge’s acceptance rate relative to the underlying picker; the M4 lift is empirical (§5.1).

Design rule 2 (§1) maps directly to this module: *auxiliary anchor-relevance signals are systematically miscalibrated against the Llama judge’s entailment surface*. Cross-encoder pickers [20] we tried (argmax-by-claim-relevance, shortest-of-top-3, earliest-of-top-5) all underperform the stub heuristic on Llama recall, because cross-encoder objectives select claim-relevant body sentences whose attribution chains and hedges Llama refuses to entail standalone. The stub’s “first non-boilerplate sentence” heuristic does slightly better by exploiting the lead-bias [21] property under which short declarative leads tend to be self-contained, but it is still an auxiliary signal, not the judge’s signal. The fix is to gate on the judge itself: ShadowVal calls the same Llama with the same prompt the scorer will use, and accepts only its YES verdicts.

ShadowVal lifts the stub by +0.0145 M4 at full val (95% CI [+0.0125, +0.0166]), entirely from precision/recall (+0.031, with entailment essentially flat). Item-level: 60% of items get at least one anchor swap and on those items P/R rises from near-zero to near-one; the remaining 40% produce byte-identical output to the underlying picker. The lever is rating-independent (see §4.5). Stacked with HostCite, ShadowVal contributes an additional +0.0075 M4 (95% CI [+0.0056, +0.0095]) — about half of its standalone effect. Both levers improve cite-sentence P/R; on items where HostCite already pushes the cite past Llama’s YES threshold ShadowVal has nothing to add, so the composition is sub-additive (stack-over-sum ≈ 0.80). The marginal lift is nonetheless confidently positive, ruling out the reading that ShadowVal rides entirely on cite-framing context.

4.5. Validation manifests (v1 and v2)

Full-validation scoring on 3372 items costs roughly 14 hours per configuration; the iteration cadence we needed (one experiment per 50–90 minutes) requires a 200-item subsample. We use two stratifications. **v1** is reviewer-stratified to match the test reviewer mix; **v2** is jointly stratified by reviewer $\times$ three-bucket rating (TRUE / FALSE / PARTIAL via `label.rating` $\in \{0, 1, 2\}$ with string-mapping fallback for unrated Snopes items). v2 was added because v1 had zero coverage of Mixture and Correct Attribution items, which are roughly 12% of test. The ratio between a lever’s v2 effect and its v1 effect diagnoses the lever’s character: ratio near 1.0 identifies rating-independent levers (ShadowVal, 1.06 $\times$), and ratio above 1.5 identifies rating-mix-sensitive levers (HostCite, 1.9 $\times$).

4.6. Implementation

The deterministic modules are plain Python; RoBERTa entailment runs on GPU via transformers [22]. All numbers in §5 are measured on a single GPU host (RTX 5090); hardware, model digests, and a ~ 0.01 – 0.02 M4 CPU/MPS-vs-CUDA absolute shift are documented in Appendix A.

Table 1

Validation M4 across measured systems. “Dev” is the v2 200-item reviewer $\times$ rating-stratified manifold used for fast iteration; “Full” is the 3372-item WatClaimCheck validation split. Bold marks our lead submission. At full val, *StatesThat+ShadowVal* (0.5243) is statistically tied with the lead submission (0.5238); see Table 2.

Submission	Anchor pick	Cite frame	Dev M4 $\uparrow$	Full M4 $\uparrow$
$K = 1$ stub	first-clean	bare	0.4819	0.4964
ShadowVal alone	shadow-validated	bare	0.4960	0.5109
HostCite	first-clean	According-to	0.4960	0.5163
StatesThat	first-clean	states-that	0.5067	0.5170
StatesThat+ShadowVal	shadow-validated	states-that	0.5048	0.5243
ShadowVal+HostCite	shadow-validated	According-to	0.5077	0.5238

Table 2

Paired bootstrap 95% confidence intervals on M4 deltas at full val ($n = 3372$). Significance flagged where the interval excludes zero. The bottom block reports the two ties: *StatesThat* and *HostCite* as cite frames, and the two top-scoring stacks.

Comparison	Δ M4	95% CI	Significant
ShadowVal+HostCite vs $K = 1$	+0.0274	[+0.0248, +0.0301]	✓
StatesThat+ShadowVal vs $K = 1$	+0.0279	[+0.0254, +0.0306]	✓
HostCite vs $K = 1$	+0.0199	[+0.0176, +0.0222]	✓
StatesThat vs $K = 1$	+0.0206	[+0.0184, +0.0229]	✓
ShadowVal alone vs $K = 1$	+0.0145	[+0.0125, +0.0166]	✓
ShadowVal+HostCite vs HostCite	+0.0075	[+0.0056, +0.0095]	✓
ShadowVal+HostCite vs ShadowVal alone	+0.0129	[+0.0105, +0.0153]	✓
StatesThat vs HostCite	+0.0007	[−0.0012, +0.0027]	— tied
ShadowVal+HostCite vs StatesThat+ShadowVal	−0.0005	[−0.0026, +0.0016]	— tied

5. Experiments and Findings

5.1. Validation scoreboard and confidence intervals

Table 1 reports M4 across measured systems on the v2 200-item development manifold and on the full 3372-item validation split. The lead submission (*ShadowVal+HostCite*) clears the $K = 1$ stub by +0.0274 M4 at full val; *HostCite* alone delivers most of that (+0.0199) and *ShadowVal* alone delivers +0.0145, but the two compose sub-additively (stack-over-sum ≈ 0.80) because both raise the same cite-sentence P/R signal on overlapping items. The v2 manifold worked as a fast-iteration proxy but had a transfer ceiling: deltas-vs- $K = 1$ on v2 predict full-val deltas directionally for every lever we measured, but absolute magnitudes shift by up to ~ 0.006 between manifolds, occasionally in directions that change tied-vs-significant verdicts. Table 2 reports paired bootstrap [23] 95% confidence intervals on the full-val deltas.

Three takeaways. First, our lead submission clears $K = 1$ by a confidently-detected margin (Table 2). Second, the two levers compose sub-additively: *ShadowVal*’s +0.0075 marginal lift on top of *HostCite* is roughly half of its +0.0145 standalone effect, indicating overlap on the cite-sentence P/R signal both levers raise. The lift is nonetheless confidently positive, ruling out the reading that *ShadowVal* rides on cite-framing context alone. Third, the *StatesThat+ShadowVal* stack — which on the v2 development manifold sat 0.003 below the lead submission — at full val ties it ($\Delta = -0.0005$, 95% CI [−0.0026, +0.0016]) while clearing $K = 1$ by +0.0279. The v2 manifold ranked *ShadowVal+HostCite* above *StatesThat+ShadowVal* by 0.003; at full val they are tied, which is the cleanest example in our matrix of why fast-iteration estimates need full-val confirmation before a system is declared dominant.

Table 3Test-set 3-metric scoreboard ($n = 1158$, no entailment because test references are sealed).

Submission	P $\uparrow$	R $\uparrow$	C $\uparrow$	mean-of-3 $\uparrow$	Δ vs $K = 1$
$K = 1$ stub	0.335	0.335	0.987	0.552	—
HostCite	0.364	0.364	0.987	0.571	+0.019
StatesThat	0.365	0.365	0.987	0.572	+0.020
ShadowVal alone	0.373	0.373	0.987	0.577	+0.025
StatesThat+ShadowVal	0.384	0.384	0.987	0.585	+0.033
ShadowVal+HostCite	0.389	0.389	0.987	0.589	+0.037

Table 4Paired bootstrap 95% CIs on the test 3-metric sub-score ($n = 1158$).

Comparison	Δ	95% CI	Significant
ShadowVal+HostCite vs $K = 1$	+0.0361	[+0.0301, +0.0424]	✓
ShadowVal+HostCite vs HostCite	+0.0173	[+0.0134, +0.0215]	✓
StatesThat+ShadowVal vs $K = 1$	+0.0323	[+0.0266, +0.0381]	✓
ShadowVal+HostCite vs StatesThat+ShadowVal	+0.0039	[−0.0009, +0.0085]	—

5.2. Test-set scoreboard (3-metric, visible)

Because the human reference articles for the test split are sealed, participants only observe the three article-side metrics (precision, recall, coverage) during the visible competition phase. Table 3 reports those, along with paired bootstrap CIs over test items ($n = 1158$) in Table 4. *Scorer-version note:* Table 3’s P/R reflects the visible/pre-reveal scorer used during the competition phase; the official post-reveal scorer (Table 7) applies a corrected citation parser and reduces our P/R to 0.299, which we discuss in §5.5.

The key result from the test scoreboard is in the last row of Table 4: *ShadowVal+HostCite and StatesThat+ShadowVal are statistically tied on the visible test sub-score.* The v2 development manifold had placed ShadowVal+HostCite 0.003 ahead of StatesThat+ShadowVal — a margin we later confirmed at full val to be noise (§5.1). The operational consequence at submission time was that we could not distinguish the two on the only signal we could observe, so both had to remain in the portfolio.

Disaggregating by reviewer (Table 5) reveals that the pooled tie hides a real per-reviewer split: StatesThat+ShadowVal significantly outperforms ShadowVal+HostCite on the AFP slice (paired bootstrap $\Delta = -0.0054$, 95% CI [−0.0109, −0.0002]), while the two are statistically indistinguishable on every other reviewer. The *According to {host}* frame matches PolitiFact and Snopes attribution patterns well; the *{host} states that* frame matches AFP’s review-attribution prose better. The strongest empirical statement we can make about portfolio composition is therefore not “ship both as a hedge” but “the AFP slice has a measurably different best-arm than the rest of test, and we ship StatesThat+ShadowVal specifically to win it.”

5.3. The third lever: StatesThat

HostCite (§4.3) and ShadowVal (§4.4) are the two levers that ship in our lead submission. A third attribution variant, *StatesThat* (wrapping each citation as *{host} states that {anchor}* (source: {url})), turned out at full val to be statistically indistinguishable from HostCite as a cite frame ($\Delta = +0.0007$, 95% CI [−0.0012, +0.0027]); the StatesThat+ShadowVal stack ties our lead submission at the top of the val scoreboard (§5.1). On the v2 development manifold this stack sat 0.003 below our lead submission, which is why ShadowVal+HostCite rather than StatesThat+ShadowVal was identified as our lead submission at competition time; both shipped in the portfolio. The two attribution patterns differ in per-reviewer preference: *According to {host}* matches PolitiFact and Snopes prose, *{host} states that* matches AFP attribution patterns better (§5.2).

Table 5

Per-reviewer test 3-metric mean across all six measured systems. Column headers: **Stub** = $K = 1$ stub, **HostC** = HostCite, **StTh** = StatesThat, **ShVal** = ShadowVal alone, **SThSV** = StatesThat+ShadowVal, **ShV+HC** = ShadowVal+HostCite (lead submission). Bold marks the per-reviewer winner. The split between StatesThat+ShadowVal (best on AFP) and ShadowVal+HostCite (best on PolitiFact, Snopes, FactCheck.org) is the empirical justification for the portfolio strategy in §6.3.

Reviewer	n	Stub	HostC	StTh	ShVal	SThSV	ShV+HC
PolitiFact	686	0.5961	0.6155	0.6183	0.6259	0.6294	0.6344
AFP Fact Check	253	0.4293	0.4401	0.4511	0.4400	0.4552	0.4498
Snopes	171	0.5739	0.6043	0.5837	0.6029	0.6181	0.6281
FactCheck.org	27	0.4729	0.4870	0.4843	0.4838	0.4857	0.4951
USA Today	21	0.5362	0.5478	0.5463	0.5639	0.5398	0.5623

5.4. Negative results: ablations that did not ship

We ran 13 ablations whose output never reached the leaderboard. Table 6 summarises them with mechanism in one line. Two kills generated the scorer constraints in §3.4 and are discussed there; the rest group into three lever families (LLM-drafting, anchor-selection refinements, reviewer-style lede injection) plus three standalone probes that the table covers self-contained.

LLM-drafting at three abstraction levels (Qwen3:8B full-body, Qwen3:30B-A3B full-body, Qwen3:30B-A3B lede-only) all lose in the same direction by the same mechanism. Full-body drafts at 8B cost -0.026 M4 vs the stub on $n = 29$; scaling to 30B widens the gap to -0.070 vs our lead submission ($n = 50$); restricting the LLM to the lede only (chunk count preserved) still costs -0.011 because the LLM’s reviewer-style lede is not entailed by the actual reference distribution. The lever family appears exhausted at our prompt-fidelity quality without a fundamentally different design (e.g., constrained decoding to enforce verbatim cite anchors).

Anchor-selection refinements (G1 per-reviewer cite frame, G2 prefer-long anchor, G3 strict noisy-fragment filter) all lose against the lead submission. G2 and G3 break the lead-bias property under which Llama recall scores short declarative leads better; G1’s apparent AFP gain is an artefact of HostCite’s combined frame swap and disappears when only the AFP slice is reverted on top of the lead submission.

Reviewer-style lede injection (G4) attempts to replace the stub lede with a nearest-neighbour reviewer lede drawn from the train split. G4 loses by -0.019 vs $K = 1$ alone and -0.024 vs the lead submission, with 22% of items hitting $P = 0$ because reviewer-page chrome (donation forms, social-share widgets, page metadata) leaks into the swapped lede and breaks downstream citation extraction. A G4-v2 with hand-curated per-site chrome filters reduces but does not fix the leakage (21.5% $P = 0$ rate, down from 22%); harder failure modes (Snopes *Example/Rating* blocks, complex-PolitiFact rating-button overflow, FactCheck.org bullet articles) require deeper structural parsing than line-pattern matching. The lever may still have headroom behind a more robust extractor; we leave that for future work.

5.5. Official competition results

The test phase closed on 2026-05-10 and the organisers released the official 4-metric leaderboard on 2026-05-14. **UTS placed 2nd of 11 teams** with M4 = 0.484, behind Outsider (0.546) and ahead of DS GT CheckThat (0.428). Table 7 reproduces the full leaderboard.

Three observations decompose where the result was earned and where it was lost. **Entailment** (0.341, ranked 2nd) is where HostCite pays back: the only team to beat us on entailment (UCPH RMT, 0.364) does so at a precision/recall cost so steep that their M4 sits 0.065 below ours, and our entailment delta over the baseline (+0.043) is in line with the HostCite lift we measured at val. **Coverage** (0.996, the highest on the leaderboard) is a precondition for top placement rather than a differentiator: three of the top five teams ship coverage ≥ 0.99 , and Outsider’s rank-1 finish with coverage 0.563 is the single anomaly among the top five. **Citation precision/recall** (0.299, ranked 6th) is where we lost: Outsider’s 0.671 P/R is more than twice ours despite our shadow-validated anchor picker, and their conjunction of

Table 6

Ablations that did not ship. “Alone” magnitudes are versus $K = 1$; “stacked” magnitudes are versus our lead submission ShadowVal+HostCite. *Note on G1*: its isolated $\Delta = +0.006$ vs $K = 1$ is positive, but the lever is classified as killed because it loses -0.015 when stacked on the lead submission and its apparent AFP slice gain is an artefact of HostCite’s combined frame swap (see prose above).

Lever	Δ M4	Mechanism (one line)
$K = 2$ multi-anchor	-0.014 alone	each extra cite is a P/R denominator without a numerator
HostCite+Body	-0.007 vs HostCite	templated mid-lede sentence; violates chunk-count invariance (§3.4)
HostCite+Reviewer	-0.020 vs HostCite	reviewer-name token in cite; violates cite-entailability (§3.4)
H2 length-padding	-0.010 alone	padded with raw evidence; added chunks are publisher-voice, not reviewer-voice
Qwen3:8B full-body draft	-0.026 vs stub	LLM-drafted chunks miss reference token patterns ($n = 29$)
Qwen3:30B-A3B full-body draft	-0.070 stacked	same as 8B at higher capacity; cite-format drift dominates ($n = 50$)
Qwen3:30B-A3B lede only	-0.011 stacked	chunk-content failure even when chunk count is preserved ($n = 50$)
G3 strict noisy-anchor filter	$-0.014 / -0.018$	over-rejects; placeholder fallback kills P/R harder than the noisy-but-keyword-rich anchor
G2 prefer-long anchor	$-0.010 / -0.015$	prefers anchors ≥ 50 chars; breaks lead-bias property that $K = 1$ wins on
G1 per-reviewer cite frame	$+0.006 / -0.015$	AFP gain is an artefact of HostCite’s combined frame swap
G4 train-ref NN lede swap	$-0.019 / -0.024$	22% of items hit $P = 0$ from reviewer-page chrome leakage
G4-v2 per-site chrome filters	$-0.023 / -0.029$	hand rules help some sites; harder failure modes need structural parsing
BGE-M3 cross-encoder rerank [24]	$+0.001 / -0.013$	rerank of ShadowVal fallback; cross-encoder picks claim-relevant compound sentences that MNLi scores worse than doc-order leads

Table 7

Official CheckThat! 2026 Task 3 leaderboard (test phase, $n = 1158$). Columns are the four shared-task metrics and their unweighted mean (M4); rows sorted by M4. “Baseline” is the organiser-provided baseline.

Rank	Team	E $\uparrow$	Cite P $\uparrow$	Cite R $\uparrow$	Cov $\uparrow$	M4 $\uparrow$
1	Outsider	0.278	0.671	0.671	0.563	0.546
2	UTS (this paper)	0.341	0.299	0.299	0.996	0.484
3	DS GT CheckThat	0.295	0.240	0.276	0.902	0.428
4	Facty	0.229	0.231	0.254	0.995	0.427
5	UCPH RMT	0.364	0.144	0.174	0.993	0.419
6	Ro	0.321	0.463	0.463	0.343	0.398
7	Team AKSH	0.329	0.301	0.318	0.564	0.378
8	Sourceminds	0.245	0.337	0.339	0.394	0.329
9	CheckMate	0.221	0.350	0.350	0.391	0.328
10	KNU Fact	0.227	0.287	0.295	0.440	0.312
11	Shtian	0.279	0.194	0.198	0.210	0.220
—	Baseline	0.298	0.223	0.240	0.329	0.272

sub-0.6 coverage with dominant P/R is the single anomalous pattern in the top of the leaderboard. We discuss its candidate mechanism — *selective emission* — and the per-cite confidence-gate experiment that would refute it, in §6.1.

Submission selection. The leaderboard rule selects each team’s single highest-scoring entry under joint four-metric evaluation. The selected entry was ShadowVal+HostCite (lead submission);

StatesThat+ShadowVal scored 0.585 on the visible test 3-metric versus 0.589 for the lead submission (paired bootstrap +0.004 for the lead-vs-alt direction, 95% CI $[-0.0009, +0.0085]$, tied within noise). The post-hoc full-val measurement reveals the two stacks tied on the 4-metric score as well ($\Delta = -0.0005$, $[-0.0026, +0.0016]$), so the leaderboard’s choice of ShadowVal+HostCite was consistent with the joint evidence but was not a clean signal at submission time. The portfolio diversification strategy in §6.3 therefore acted as a defensive hedge: it did not change the leaderboard outcome (the best-of-N rule means only the lead entry matters), but it provided robustness against the val→test transfer uncertainty we had no way to measure pre-reveal.

5.6. Operational findings

The v1-vs-v2 lever-effect ratio diagnoses lever character (rating-independent vs rating-mix-sensitive) before submission, and would have flagged HostCite’s sensitivity earlier had v2 existed from the start. Separately, citation-format hygiene is a hard prerequisite: the IFM_MBZUAI K2 baseline fails the citation-format regex on 5/20 probe items, zeroing those items’ P/R, and any new team must clear that bar before any other lever pays off. Reproducibility caveats (scorer version skew, CPU/MPS-vs-CUDA shift, model digests) are in Appendix A.

6. Discussion

6.1. The headline gap: precision/recall to the winner

Reading Table 7 along the columns rather than the rows isolates what determined our placement. Our P/R deficit to Outsider on the official scorer (0.299 vs 0.671) outweighs our wins on entailment (+0.063) and coverage (+0.433), leaving our 4-metric average 0.062 below Outsider’s. Every lever in this paper edits cite *content* (HostCite, ShadowVal, StatesThat, the portfolio strategy); on the visible-scorer test set the full stack lifts P/R from 0.335 ($K = 1$ stub) to 0.389, a ceiling well below Outsider’s 0.671 regardless of which scorer the comparison projects onto.

We hypothesise the missing axis is *selective emission*: top-ranked systems likely decline to emit a cite when the per-cite faithfulness judge would refuse, whereas our system emits one cite per evidence URL unconditionally. Outsider’s coverage backs this from their side: at 0.563 they are the only top-five team below 0.99, exactly the pattern a drop-on-low-confidence policy produces — fewer URLs cited, but each surviving cite passes the judge. Our own matrix backs it from the opposite direction: the three independent cite-content levers — frame text (HostCite, StatesThat) and anchor selection (ShadowVal) — each plateau at roughly +0.04 visible-scorer test P/R over $K = 1$ and stack only sub-additively (§5.1), so the cite-content lever family is near its ceiling on our pipeline. Changing *which* cites are emitted, not *how* they read, is the natural remaining axis.

We did not explore selective emission during the campaign. We had read the cite-entailability constraint (§3.4) as a one-way rule — every emitted cite must be entailable — and missed its corollary: *some candidates may be better dropped than emitted*, and that policy lever is what closes the P/R gap. The targeted refutation we recommend is a per-cite confidence gate: run the ShadowVal judge on the chosen anchor and *drop the cite entirely* on NO, rather than escalating to the next candidate. Holding the rest of the pipeline fixed, this isolates emission policy from anchor selection and would resolve whether selective emission alone closes the gap, or whether the top systems combine it with a categorically better anchor selector. The hypothesis is consistent with the leaderboard pattern but not directly tested in our matrix; the per-cite gate is the experiment we would run first in the next cycle.

6.2. What generalises beyond CheckThat!

Chunked NLI scoring against a held-out reference is becoming a common pattern in generation benchmarks where bounded-domain accuracy is hard to measure end-to-end [25]. Wherever the scorer chunks both the generation and the reference, the chunk-count-invariance constraint (§3.4) should transfer;

templated changes that enrich existing chunks beat templated changes that add or rewrite chunks. Llama-3.2:1B as a per-cite judge is generous and surface-pattern-driven, which means short literal cites win against long evidence premises and any pipeline that paraphrases “to improve readability” is gambling against the judge; this transfers to any benchmark where a small open language model scores citation faithfulness. Finally, the LLM-drafting failure we report — three independent attempts, same mechanism — is a reproducible negative result that anyone evaluating LLM-as-drafter on chunked-NLI metrics should expect; the plausible fix (constrained decoding to enforce verbatim cite anchors) is unverified. These transfers concern scoring mechanics, not writing quality: the deterministic templated drafting that wins under this scorer does not produce the fluent long-form prose automated fact-check journalism ultimately needs, and on that axis agentic LLM drafting [16] — still lagging expert-written articles in human evaluation — remains the relevant research direction.

6.3. Portfolio strategy under no-live-feedback

The competition offers a generous submission budget under best-of-N scoring with no per-submission feedback during the visible phase. Those rules favour spreading submissions over refining one: rather than push the top-val candidate further, cover the cells where val→test transfer is uncertain. Our six-entry portfolio covers the (cite-frame text × anchor-pick method) cells. The diversification did not change our leaderboard placement — under best-of-N only the single best entry counts — but it provided robustness against transfer noise we had no way to measure pre-reveal, and the per-reviewer split in Table 5 shows a real underlying signal: the AFP slice has a measurably different best-arm (StatesThat+ShadowVal) than the rest of test (ShadowVal+HostCite), and the cite frames that drive the split are themselves statistically tied at the full-val pool level.

6.4. Future work and what we would do differently

The top item for the next cycle is the per-cite confidence gate that tests the selective-emission hypothesis (§6.1): drop low-confidence cites instead of emitting them, isolating emission policy from anchor selection. Beyond that, three calibrations stand out in hindsight. First, build the v2 rating-stratified manifold before running v1-tuned ablations; this would have flagged HostCite’s rating-mix sensitivity earlier and likely shortened the cite-frame sweep by half. Second, enforce the chunk-count and cite-entailability invariants programmatically rather than by running −0.01 M4 experiments to rediscover them; both can be expressed as one-line predicates over the rendered article. Third, for the LLM-drafting family, try constrained decoding [6] to enforce verbatim cite anchors before writing the family off. That is the genuine refutation of the negative result we report.

7. Conclusion

UTS placed 2nd of 11 teams on the official CheckThat! 2026 Task 3 leaderboard ($M4 = 0.484$); the +0.062 gap to the winner sits entirely on citation precision/recall (0.299 vs 0.671), while entailment (2nd) and coverage (tied 1st) are at or near the top (§6.1). The placement was earned by a deterministic stub generator wrapped by two single-lever interventions deriving from two design rules: *scorer conservatism* (both sub-judges credit only what they can verify against their references) and *auxiliary anchor signals are miscalibrated against the judge* (cross-encoder, length, and lead-position heuristics pick anchors the judge refuses to entail). The interventions — HostCite, a domain-attribution cite frame, and ShadowVal, an anchor picker that mirrors the Llama judge — lift $M4$ by +0.0274 on the full 3372-item validation split (paired bootstrap, 95% CI [+0.0248, +0.0301]), composing sub-additively because both raise cite-sentence P/R on overlapping items. The two design rules transfer beyond CheckThat!: any pipeline using a small open language model as a per-cite faithfulness judge should expect anchor-selection auxiliary signals to be miscalibrated, and any pipeline scored under chunked-NLI against a held-out reference should expect that emitting content the judge cannot verify drags both sub-metrics down. The lever we did not test, and recommend first for the next cycle, is *selective emission*: a per-cite confidence

gate that drops the cite when the judge would refuse. We hypothesise this is the mechanism behind the leaderboard’s top entries that trade coverage for P/R. Constrained decoding for LLM-drafted bodies (§6) is a second direction worth trying before writing off that lever family.

Acknowledgments

We thank the CheckThat! organisers for their responsiveness in patching the scorer mid-campaign and for keeping the leaderboard infrastructure stable through the scorer-version updates. We thank the developers of `ceurart`, `ollama`, and the `transformers` library for tooling the experiment matrix relied on.

Declaration on Generative AI

During the preparation of this work, the author(s) used Qwen3:8B and Qwen3:30B-A3B (locally served) as the subjects of ablations described in §5.4. The author(s) also used Claude (Anthropic) for editorial assistance on the prose; the author(s) reviewed and approved all content and take full responsibility for the paper.

References

- [1] J. M. Struß, S. Schellhammer, S. Dietze, V. Venkatesh, V. Setty, T. Chakraborty, P. Nakov, A. Anand, P. Chungkham, S. Hafid, D. Sahnan, K. Todorov, The CLEF-2026 CheckThat! lab: Advancing multilingual fact-checking, in: *Advances in Information Retrieval. Proceedings of the 48th European Conference on Information Retrieval (ECIR 2026)*, Lecture Notes in Computer Science, Springer, Cham, 2026. doi:10.1007/978-3-032-21321-1_43. arXiv:2602.09516.
- [2] D. Sahnan, T. Chakraborty, P. Nakov, Overview of the CLEF-2026 CheckThat! lab task 3 on generating full fact-checking articles, in: *Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, CEUR-WS.org, 2026. Forthcoming.
- [3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019. arXiv:1907.11692.
- [4] A. Williams, N. Nangia, S. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1112–1122. URL: <https://aclanthology.org/N18-1101/>. doi:10.18653/v1/N18-1101.
- [5] Qwen Team, Qwen3 technical report, <https://github.com/QwenLM/Qwen3>, 2025.
- [6] B. T. Willard, R. Louf, Efficient guided generation for large language models, 2023. arXiv:2307.09702, implementation: Outlines (<https://github.com/dottxt-ai/outlines>).
- [7] Meta AI, The Llama 3 herd of models, <https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/>, 2024. Llama 3.2 release; 1B and 3B text models.
- [8] K. Khan, R. Wang, P. Poupart, WatClaimCheck: A new dataset for claim entailment and inference, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 1293–1304. URL: <https://aclanthology.org/2022.acl-long.92/>. doi:10.18653/v1/2022.acl-long.92.
- [9] LLM360, K2-V2-instruct: An instruction-tuned open reasoning model, <https://huggingface.co/LLM360/K2-Chat>, 2025. External baseline submission to CheckThat! 2026 Task 3.
- [10] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: A large-scale dataset for fact extraction and VERification, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long*

- Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. URL: <https://aclanthology.org/N18-1074/>. doi:10.18653/v1/N18-1074.
- [11] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, *Transactions of the Association for Computational Linguistics* 10 (2022) 178–206. URL: <https://aclanthology.org/2022.tacl-1.11/>. doi:10.1162/tacl_a_00454.
 - [12] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, P. Przybyła, T. Caselli, G. Da San Martino, F. Haouari, M. Hasanain, C. Li, J. Piskorski, F. Ruggeri, X. Song, R. Suwaileh, Overview of the CLEF-2024 CheckThat! lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness, in: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 15th International Conference of the CLEF Association (CLEF 2024)*, volume 14959 of *Lecture Notes in Computer Science*, Springer, Cham, 2024, pp. 28–52. doi:10.1007/978-3-031-71908-0_2.
 - [13] P. Atanasova, J. G. Simonsen, C. Lioma, I. Augenstein, Generating fact checking explanations, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020, pp. 7352–7364. URL: <https://aclanthology.org/2020.acl-main.656/>. doi:10.18653/v1/2020.acl-main.656.
 - [14] N. Kotonya, F. Toni, Explainable automated fact-checking for public health claims, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 7740–7754. URL: <https://aclanthology.org/2020.emnlp-main.623/>. doi:10.18653/v1/2020.emnlp-main.623.
 - [15] F. Zeng, W. Gao, JustiLM: Few-shot justification generation for explainable fact-checking of real-world claims, *Transactions of the Association for Computational Linguistics* 12 (2024) 334–354. URL: <https://aclanthology.org/2024.tacl-1.19/>. doi:10.1162/tacl_a_00649.
 - [16] D. Sahnan, D. Corney, I. Larraz, G. Zagni, R. Miguez, Z. Xie, I. Gurevych, E. Churchill, T. Chakraborty, P. Nakov, Can LLMs automate fact-checking article writing?, *Transactions of the Association for Computational Linguistics* (2026). arXiv:2503.17684, to appear.
 - [17] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, C. Zhu, G-Eval: NLG evaluation using GPT-4 with better human alignment, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 2511–2522. URL: <https://aclanthology.org/2023.emnlp-main.153/>. doi:10.18653/v1/2023.emnlp-main.153.
 - [18] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL: <https://openreview.net/forum?id=SkeHuCVFDr>.
 - [19] Ollama Team, Ollama: Get up and running with large language models locally, <https://ollama.com>, 2024.
 - [20] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: <https://aclanthology.org/D19-1410/>. doi:10.18653/v1/D19-1410.
 - [21] C. Kedzie, K. McKeown, H. Daumé III, Content selection in deep learning models of summarization, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1818–1828. URL: <https://aclanthology.org/D18-1208/>. doi:10.18653/v1/D18-1208.
 - [22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2020, pp. 38–45. URL: <https://aclanthology.org/2020.emnlp-demos.6/>. doi:10.18653/v1/2020.emnlp-demos.6.
 - [23] P. Koehn, Statistical significance tests for machine translation evaluation, in: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Association for

Computational Linguistics, Barcelona, Spain, 2004, pp. 388–395. URL: <https://aclanthology.org/W04-3250/>.

- [24] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, in: Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 2318–2335. URL: <https://aclanthology.org/2024.findings-acl.137/>. doi:10.18653/v1/2024.findings-acl.137.
- [25] Y. Zha, Y. Yang, R. Li, Z. Hu, AlignScore: Evaluating factual consistency with a unified alignment function, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11328–11348. URL: <https://aclanthology.org/2023.acl-long.634/>. doi:10.18653/v1/2023.acl-long.634.

A. Reproducibility

Code and data. The system reproduces from source repository commit 2744d3f (or any descendant). Manifest construction scripts (v1 reviewer-stratified, v2 reviewer $\times$ rating-stratified, full 3372-item) live in `system/manifests/`. The lab scorer pinning is the `c1ef2026-checkthat-lab-main` 2026-04-27 snapshot.

Models. The Llama-3.2:1B model blob digest used for scoring and ShadowVal shadow validation is `sha256:74701a8...8d45`; we pin it because at `temperature=0` the model is byte-deterministic on our hardware, and re-pulling the latest tag without verifying the digest can silently shift absolute scores by 0.005–0.01 M4. Citation parsing uses the scorer’s current regex (`;-` separated URLs only).

Hardware and runtime. Development on an M-series MacBook Pro; production runs on a local Ubuntu host (RTX 5090, 24 GB). End-to-end GPU-host runtime is ~ 50 –90 min per 200-item validation pass, ~ 14 h per full-val pass, and ~ 5 h for the 1158-item test split. Local CPU/MPS Llama and CUDA Llama produce a systematic ~ 0.01 –0.02 M4 absolute shift (rankings preserved); single-host comparisons are the only safe unit, so we re-baseline on the GPU host before reporting deltas.