ReviewMap: Mapping Evaluative Acts and Automation Boundaries in Peer-Review Comments

Anonymous ACL submission

Abstract

Peer-review comments check what a paper does and judge why it matters, but no existing scheme distinguishes these two evaluative acts. We argue that operationalizing this distinction and measuring where LLMs can reliably do so defines a more honest mission for NLP in peer review than full automation, and is a prerequisite for responsible LLM use in the process. We present **ReviewMap**, a framework that classifies each comment as *assessment* or *calibration*, with finer-grained attributes applied conditionally by act type. Applied to 500 ICLR 2026 papers through multi-model LLM consensus, the act layer proves reliably automatable (inter-model $\kappa \geq 0.75$) while the attribute layer requires human oversight, defining a measured automation boundary. At the corpus level, we find that 17% of reviewer–paper instances contain no explicit calibration, and calibration concentrates roughly 3× more in Strengths than in Weaknesses. Finally, we derive three affordances from this boundary: a review guideline that operationalizes the assessment-then-calibration workflow for reviewers, a per-comment rebuttal strategy scaffold for authors, and a discussion-phase dashboard for area chairs.

1 Introduction

Peer review requires evaluating a submission from multiple angles: whether claims are supported by adequate evidence, whether the methodology meets the standards of the field, and whether the work constitutes a contribution significant enough for the venue. These dimensions are often recorded through separate criterion scores but ultimately inform an overall recommendation. Yet they differ in kind. Checking whether an experiment lacks a proper baseline is a different evaluative act from declaring that the contribution is too incremental. The first verifies what the paper does against shared standards; the second commits a judgment about why the work matters. We call these two acts *assessment* and *calibration*. Venue reviewer guidelines already encode a related separation at the scoring-criterion level: ACL Rolling Review separates soundness, excitement, and overall recommendation; ICML separates soundness from significance; and NeurIPS distinguishes quality from significance. ICLR, the venue our corpus draws from, instead uses a *contribution* criterion that can combine novelty, value, and venue-level significance. Across venues, however, the free-text comments interleave these evaluative acts without marking them as structurally distinct.

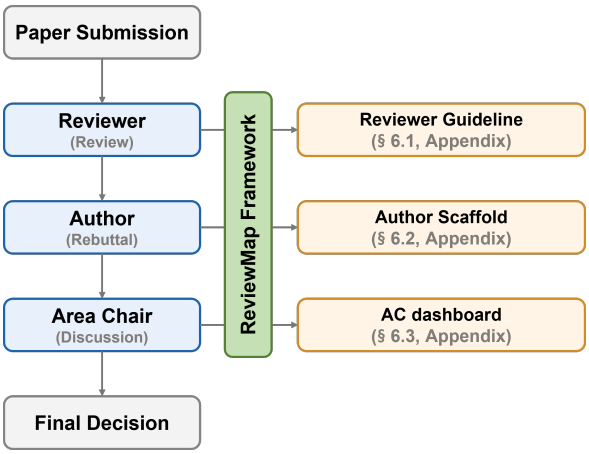

Figure 1: The ReviewMap ecosystem. The peer-review timeline (left) routes through three role engagements; a single comment-level coding scheme (center) feeds each role one affordance (right; §6.1, §6.2, §6.3). What each affordance computes autonomously follows the automation boundary measured in §5.

The growing volume of submissions has motivated NLP-for-peer-review research (Kuznetsov et al., 2024) and raised concerns about the sustainability of the AI-venue peer-review process at scale (Kim et al., 2025). Recent work has begun deploying LLMs in this process, from detecting LLM-modified reviews (Liang et al., 2024) to providing automated feedback to reviewers at scale (Thakkar

et al., 2026) and simulating peer review dynamics with LLM agents (Jin et al., 2024). However, these efforts operate without a clear framework for determining which parts of review analysis LLMs can reliably perform and which require human judgment. Baumann et al. (2026) argue that deploying LLMs in peer review without rigorous evaluation of their reliability risks undermining the process. Yet no existing scheme provides the measurement instrument such evaluation would require.

Prior schemes classify comments by topic, pragmatic action, or quality attributes; none distinguishes the evaluative act itself. A comment about novelty can be either factual—"The approach was previously proposed by X et al."—or a value commitment—"The novelty is insufficient for this venue." Topic and evaluative act are orthogonal.

This decomposition matters because the two acts demand different responses. Assessments can be answered with evidence: a missing experiment can be supplied, an unfair comparison rerun. Calibrations can only be engaged through argumentation: the author must reframe the contribution's significance. When a review contains no calibration, the score lacks textual justification; and when calibration is stated without explicit grounding, the judgment cannot be meaningfully engaged with.

Operationalizing this distinction and measuring where LLMs can reliably classify it is a prerequisite for responsible LLM use in peer-review analysis. We present **ReviewMap**, a framework that classifies each review comment by its evaluative act, with act-conditional attributes for finer-grained properties. It anchors an ecosystem of three role-specific affordances (Figure 1). Our contributions are: (1) **An act-based framework for peer-review evaluation.** ReviewMap distinguishes two evaluative acts at the comment level: *assessment* (what the paper does) and *calibration* (why those findings matter for the venue). (2) **A measured automation boundary for LLM-assisted review analysis.** Applying ReviewMap to 500 ICLR 2026 submissions (17,654 comments) via multi-model LLM consensus, we find the act layer reliably automatable and the attribute layer model-bound. (3) **An ecosystem translation of the same coding scheme.** We derive three boundary-respecting affordance sketches: a reviewer guideline, an author scaffold, and an AC dashboard—using the reliable act layer and treating attributes as advisory.

2 Related Work

Table 1 situates ReviewMap against prior peer-review comment analysis schemes, grouped by what they classify; we walk through each tradition below.

Aspect classification. PeerRead (Kang et al., 2018) introduced aspect-level scores from venue reviewer forms; ASAP-Review (Yuan et al., 2022) extended this to sentence-level aspect labeling; Lu et al. (2025) proposed a bottom-up, LLM-driven taxonomy with 16 coarse and 39 fine-grained aspect categories. These schemes classify what a comment is *about*. ReviewMap classifies what a comment *does*.

Pragmatic action. AMPERE (Hua et al., 2019) introduced proposition-level labels for review argumentation. DISAPERE (Kennard et al., 2022) synthesized prior schemes into sentence-level properties including Review-Action, where "Evaluative" is a single category. NLPeer (Dycke et al., 2023) contributed a multi-domain ethically licensed corpus with pragmatic tags. These schemes classify *which discourse role* a comment plays (evaluate, request, recap, . . .); DISAPERE is our closest structural precedent. ReviewMap operates one layer deeper: we decompose DISAPERE's single "Evaluative" role into assessment and calibration, and replace human annotation with multi-model LLM consensus for scalability.

Review quality attributes. SubstanReview (Guo et al., 2023) measures claim substantiation in 550 reviews, reporting a declining substantiation trend across NLP conferences over time. PolitePEER (Bharti et al., 2024) annotates peer-review comments by politeness intensity across five levels. LazyReview (Purkayastha et al., 2025) derives 18 lazy-thinking categories from ARR guidelines and validates them through a controlled experiment. These schemes apply quality attributes uniformly across all comments. ReviewMap first separates evaluative acts, then applies act-specific attributes.

Multi-model consensus. Single-model LLM annotation is effective but contested in reliability (Gilardi et al., 2023; Bavaresco et al., 2025). Verga et al. (2024) show that a panel of diverse LLMs outperforms any single judge for generation evaluation. Courvoisier et al. (2026) apply multi-model agreement to classification and extraction tasks, introducing a withheld-judgment protocol for cases

Paper	Venue	Unit	Classifies	LLM annot.	Audience
PeerRead (Kang et al., 2018)	NAACL '18	paper	Aspect scores	No	–
ASAP-Review (Yuan et al., 2022)	JAIR '22	sentence	Aspect labels	No	–
Lu et al. (Lu et al., 2025)	EMNLP F'25	sentence	Aspect (coarse/fine)	Single	R
AMPERE (Hua et al., 2019)	NAACL '19	proposition	Proposition type	No	–
DISAPERE (Kennard et al., 2022)	NAACL '22	sentence	Review-Action (+attributes)	No	AC
NLPeer (Dycke et al., 2023)	ACL '23	sentence	Pragmatic tag	No	–
SubstanReview (Guo et al., 2023)	EMNLP '23	own segment	Substantiation	No	–
PolitePEER (Bharti et al., 2024)	LRE	sentence	Politeness (5 levels)	No	–
LazyReview (Purkayastha et al., 2025)	ACL '25	own segment	Lazy thinking	Single	R
ReviewMap (ours)	—	**comment**	**Evaluative act (ASM/CAL)**	**Multi**	**R / A / AC**

Table 1: Comparison of peer-review comment analysis schemes. Prior work classifies comments by topic (aspect), pragmatic action (discourse), or quality attributes; none distinguishes the evaluative act (assessment vs. calibration). "LLM annot.": No / Single model / Multi-model consensus. "Audience": R = reviewer, A = author, AC = area chair.

without consensus. Wang et al. (2023) provide the theoretical basis for self-consistency sampling. We extend panel consensus to multi-layer subjective annotation; reliability stratifies by layer.

3 The ReviewMap Framework

A reviewer's comment performs an *evaluative act*: it characterizes what the paper does (**assessment**), commits to a judgment of its significance (**calibration**), or does neither (**off-task**). We propose **ReviewMap**: act classification at Level 1 (§3.1) and act-conditional attributes at Level 2 (§3.2). Act-conditional attributes apply only where they are semantically meaningful (claim/standard relevance for assessments, evidential grounding for calibrations); this design exposes a reliability gradient (§5) and grounds the response-strategy mapping in §6.

3.1 Level 1: Evaluative Act (ASM / CAL)

Each atomic comment in our corpus is assigned exactly one of three act codes (Table 2, top). The boundary between ASM and CAL is the locus of *value commitment*: ASM stays at the level of description (what the paper does, what its components are, what observations would clarify it), while CAL commits the reviewer to a judgment of merit. FLAG is reserved for comments that resist this distinction—pointers to other comments, file references, and non-evaluative meta-text.

3.2 Level 2: Act-conditional attributes

The second layer attaches binary attributes conditioned on the Level 1 act (Table 2, bottom). For ASM, two attributes record what the assessment evaluates against: a specific claim made by the paper (claim_relevant, P1) and/or a field-wide standard or convention (standard_relevant, P2). For CAL, a single attribute records whether the reviewer's value judgment is anchored in explicit evidence (grounded), following the substantiation literature (Guo et al., 2023). FLAG comments carry no Level 2 attributes.

Code / Attribute	Definition	Example
LEVEL 1 — EVALUATIVE ACT		
ASM (assessment)	Describes what the paper does or how it behaves.	Ablations don't report variance
CAL (calibration)	Commits to a judgment about the paper's significance.	The empirical gains are marginal
FLAG	Off-task or non-evaluative meta-text.	See major weaknesses above
LEVEL 2 — ACT-CONDITIONAL ATTRIBUTES		
claim_relevant (P1, for ASM)	Evaluates against a paper-specific claim.	Paper claims A, but Table shows B.
standard_relevant (P2, for ASM)	Evaluates against a field-wide standard.	Ablations should report variance.
grounded (for CAL)	Value judgment anchored in explicit evidence.	Novelty is limited; the derivation resembles prior work.

Table 2: The ReviewMap framework. Level 1 classifies each comment by its evaluative act; Level 2 attaches binary attributes conditioned on the Level 1 act. Full operational definitions in Appendix C; corpus distribution and reliability in §5.

P1 and P2 are *orthogonal dimensions*: a comment can score yes on both when a specific claim also conflicts with a field standard. P2 and *grounded* target adjacent but orthogonal axes: P2 is a *targeting* property (does the comment invoke a shared community standard?), while *grounded* is a *justification* property (does the comment supply

explicit evidence for a personal judgment?).

4 Methodology

4.1 Data collection

We sample 500 ICLR 2026 submissions from OpenReview (Soergel et al., 2013), stratified by primary area (21 areas) crossed with accept/reject decision; seed 42, accept rate preserved (38.8%), minimum three accepted and three rejected papers per area. Unlike aspect-level taxonomies whose categories may vary across subfields (Lu et al., 2025), the ReviewMap scheme is not tied to a single research field, so we sample across all 21 areas to examine the framework's coverage across diverse fields.

Each review is segmented into atomic comments by a rule-based parser that respects the venue's Strengths / Weaknesses / Questions template. Segmentation is conservative: nested bullets become separate comments, but adjacent text within a single bullet stays together. The 500 papers yield 17,654 atomic comments across 1,955 reviewer–paper instances. This is the unit of annotation; everything downstream (§5, §6) operates over these comments rather than over full reviews.

Separately, before the main run, we compiled an additional development set (12 papers, 470 atomic comments). The dev set is used to compare local-LLM, frontier-LLM, and human annotations before scaling to the 500-paper main corpus (§4.4).

4.2 Annotation design

We developed a single annotation guideline applied uniformly across all models (see Appendix C for the full text and worked examples; §3 summarizes the operational definitions). Each prompt joins a single paper's segmented comments with the guideline; the LLM returns one structured row per comment with the Level-1 code and the applicable Level-2 attributes.

Score-blind by design. Prompts contain only the comment text, the segmenting section header (Strengths / Weaknesses / Questions), and the paper's abstract. We deliberately withhold reviewer scores, confidence, soundness, and decision from the model. This prevents the most common leakage failure mode in score-aware annotation—model outputs that retrofit codings to the visible score—and lets us treat the LLM's coding as an independent signal that can later be cross-checked against scores.

Model	Group	Active params
Qwen 3.5	Alibaba	27 B (dense)
Gemma 4	Google	31 B (dense)
Nemotron 3-Super	NVIDIA	12 B / 120 B (MoE)

Table 3: The three open-weight local LLMs used for the corpus annotation, spanning three research groups.

Inference. We use vLLM (Kwon et al., 2023) at temperature 0.6 for self-consistency sampling (Wang et al., 2023). The 3-model × 3-run pipeline on the 500-paper main set consumes approximately 350 GPU-hours across two hardware configurations; all serving and decoding configurations are reproducible from the released artifact (Appendix A, Table 8).

4.3 Multi-model consensus design

We select three recent open-weight LLMs for cross-family diversity (Qwen 3.5 (Team, 2026), Gemma 4 (DeepMind, 2026), and Nemotron 3-Super (Chandiramani et al., 2026); Table 3). The three models come from different research groups and span dense-Transformer and Mixture-of-Experts (MoE) Mamba–Transformer architectures, allowing majority voting to reflect the underlying labeling decision rather than shared pretraining artifacts.

The annotation pipeline runs each model *three times* and resolves to a per-model coding via intra-model majority vote (Stage 1). The three per-model codings are then resolved to a final consensus via inter-model majority (Stage 2). The two-stage structure separates *stochastic* noise (Stage 1; same model, run-to-run variation) from *calibration* noise (Stage 2; different models' different priors). Figure 2 sketches the full pipeline.

Gating rule. The act code (ASM / CAL / FLAG) is resolved by 3-model majority: if at least 2 of 3 models agree, the consensus is that label. Level-2 attributes are resolved at the slot level using a 3-tier agreement scheme in the spirit of Courvoisier et al. (2026): (i) *unanimous* attribute votes auto-confirm the value; (ii) a *2-of-3 majority* confirms the value but flags the dissent for downstream readers; (iii) *no majority* records NA, surfacing the slot as a direct human-in-the-loop (HITL) item rather than forcing a value. NA values surface only on attribute fields, never on the act code, mirroring the framework's two-layer structure. Table 5 reports the per-attribute tier distribution; unanimous-vote

rates decrease along the objectivity gradient (§5.1).

Unlike prior schemes that train a dedicated classifier on labeled examples (Kennard et al., 2022; Purkayastha et al., 2025), ReviewMap uses LLMs directly as annotators against a written guideline, eliminating the train/test split and making schema iteration possible without retraining.

4.4 Pipeline validation on the dev set

Before applying the pipeline to the main corpus, we validate it on the development set against two stronger reference annotators.

Human anchor. One ML-expert human annotator (an author of this paper, blind to all LLM outputs throughout the annotation pass) codes each of the 470 dev-set comments under the same guideline using the same per-paper sheet format the LLMs see. We use this as an expert alignment anchor, not a human gold standard: it mirrors the area-chair condition of one expert reading cross-domain ICLR papers. Single-annotator coding is itself a limitation we discuss in §7; in the absence of multi-annotator agreement at this stage, the LLM-pipeline validation we report is necessarily relative to one annotator's judgments rather than to a corpus-wide gold standard.

Frontier-LLM anchor. We additionally collect a 3-frontier-model consensus on the same dev set papers via manual chat-UI sessions: three frontier models (ChatGPT 5.5 (OpenAI, 2026), Gemini Pro 3.1 (Google DeepMind, 2025), Claude Opus 4.6 (Anthropic, 2026)), three independent runs per (paper, model) (108 total chat sessions), parsed back into structured codings and majority-voted in the same 2-stage manner as the local-LLM pipeline. The frontier-LLM consensus is independent of the local-LLM pipeline (different models, different inference path, manual rather than API).

Three-tier validation result. Across the dev set we report agreement at three levels using Cohen's κ (Cohen, 1960), interpreted via Landis & Koch thresholds (Landis and Koch, 1977): Human↔Frontier ($\kappa = 0.65$, Substantial), Human↔Local ($\kappa = 0.62$, Substantial), and Frontier↔Local ($\kappa = 0.84$, Almost Perfect). The qualitative pattern across all three pairings is the same: Substantial-or-better agreement at the act layer. The open-weight pipeline is calibrated against both a human and a frontier reference on the dev set; the 500-paper main run scales the validated configuration.

Metric	Value
Intra-κ (Q / G / N)	0.84 / 0.96 / 0.78
Inter-κ (Q↔G / Q↔N / G↔N)	0.77 / 0.77 / 0.75
Act distribution % (ASM/CAL/FLAG)	80.5 / 17.4 / 2.2
Consensus split % (unan./2-of-3/no-maj)	88.4 / 11.3 / 0.3

Table 4: Act-layer reliability across the three local models (main set, $n = 17{,}654$ comments). Both intra- and inter-model agreement are Substantial or higher, with the act distribution converging tightly across models.

4.5 Scaling to the 500-paper main set

With the pipeline validated on the dev set, we apply the same configuration to all 500 papers. The result is 100% coverage on the act code: all 17,654 expected comments receive a consensus code (no missing rows). Per-model `valid%` for the act code plus required-slot attributes is Qwen 98.84%, Gemma 96.67%, Nemotron 99.76%.

Hallucinated extra rows (e.g., an invented Q4 when the reviewer wrote only Q1–Q3) are auto-filtered at the consensus join: only rows whose (paper, reviewer, comment_id) appears in the expected set enter consensus. The detailed per-model breakdown—hallucination counts and schema-violation directionality—is reported in §5.2.

5 Results

5.1 What ReviewMap reveals

Act-layer classification is reliable. The Level-1 act code is reliably classified by all three open-weight local LLMs (Table 4). Intra-model 3-run self-consistency κ is Substantial-to-Almost-Perfect for every model; inter-model pairwise κ is uniformly Substantial across all three model pairs. Distributions converge tightly—ASM 80.5% / CAL 17.4% / FLAG 2.2%—and the three-way consensus achieves 88.4% **unanimous,** 11.3% **2-of-3,** 0.3% **no-majority**.

Where reliability breaks: the objectivity gradient. The Level-2 attributes (P1 / P2 / grounded) tell a different story. Inter-model κ decreases monotonically from the act code through P1 and P2 to grounded (Figure 3)—the *objectivity gradient*: κ falls from Substantial at the act layer to Fair at *grounded*. Intra-model κ on the Level-2 attributes, however, remains Almost Perfect for Gemma (0.86–0.89), falls from Substantial to Moderate for Qwen

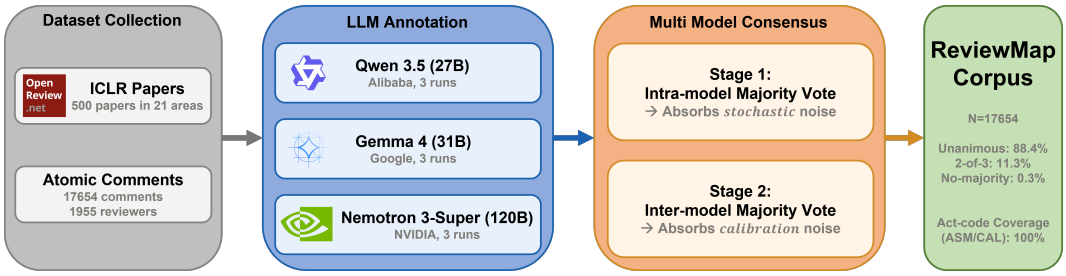

Figure 2: The 2-stage voting pipeline (§4.3). Stratified sampling produces 17,654 atomic comments from 500 ICLR 2026 papers (§4.1). A score-blind per-paper prompt (§4.2) joins the paper's comments with the annotation guideline (Appendix C). *Stage 1* resolves per-model codings via intra-model majority, absorbing run-to-run stochastic noise; *Stage 2* resolves the released consensus via inter-model majority, absorbing model-bound calibration noise. The act code uses a 3-way majority with a fixed tie-break order; Level-2 attributes use 2-of-3 confirmation and surface no-majority items as NA for human-in-the-loop adjudication.

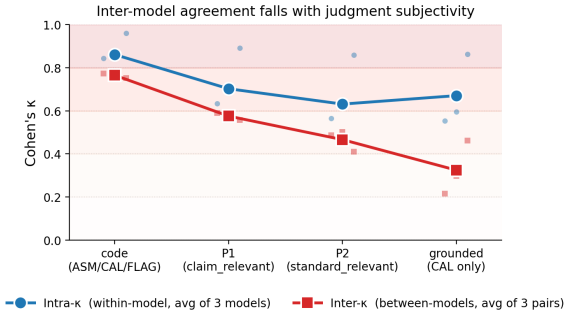

Figure 3: The objectivity gradient. Per-attribute reliability κ across the four Level-1+2 fields (act $\rightarrow$ P1 $\rightarrow$ P2 $\rightarrow$ *grounded*). Blue circles: intra-model SC κ (avg). Red squares: inter-model pairwise κ (avg). Light dots: per-model / per-pair values. Inter-model κ falls monotonically from Substantial to Fair.

Attribute (cond. act)	n	Unanim.	2-of-3	No
`claim_relevant` (ASM)	14,203	65.3	31.9	2.8
`standard_relevant` (ASM)	14,203	65.3	31.5	3.2
`grounded` (CAL)	3,067	55.5	37.5	7.0

Table 5: Attribute-layer agreement tiers (%) conditional on the relevant act-code consensus. *Unanim.* = all relevant voters agreed (auto-confirmed); *2-of-3* = majority confirmed with dissent recorded; *No* = no majority, surfaced as NA. Unanimous rates fall from 65% on the assessment attributes to 55.5% on *grounded*, mirroring the objectivity gradient (Figure 3, §5.1).

Observation	Value
Act distribution (ASM / CAL / FLAG)	80.5 / 17.4 / 2.2 %
Zero-CAL reviewer–paper instances	333/1,955 (17.0%)
Per-reviewer CAL (median / mean)	1 / 1.57
reviewers with $\leq$ 2 CALs	81.4%
ASM / CAL / FLAG share by section (%)	
Strengths	65.1 / 34.5 / 0.4
Weaknesses	85.5 / 12.3 / 2.2
Questions	89.2 / 6.9 / 3.9

Table 6: Corpus-level findings from the 3-model consensus codings. CAL is sparse and concentrated in a small subset of reviewers; calibration is markedly asymmetric across sections.

($0.63 \rightarrow 0.55$), and sits in the Moderate band for Nemotron (0.46–0.59). We treat this pattern as *model-bound* calibration: each model is more self-consistent than the panel agrees, with each model committing to its own threshold, so the attribute layer is decision-space scaffolding rather than authoritative tags (§6.4). The same pattern appears in the panel's unanimity rate (Table 5): unanimous attribute votes drop from 65% on the assessment slots to 55.5% on *grounded*, with the HITL surface widening from $\sim$ 3% to 7%.

Explicit calibration is sparse in review text. Only 17.4% of comments are CAL (explicit value commitments), and the remainder are descriptive assessments or off-task. At the review level, **17.0%** of reviewer–paper instances contain *no* CAL at all; the median reviewer issues a single CAL per paper (mean 1.57; 81.4% issue $\leq$ 2). Roughly one in six ICLR 2026 reviews are entirely descriptive, with the evaluative judgment implicit in the score rather than written in the comment text.

Strengths are calibration-heavy; Weaknesses are descriptive. Where CAL does appear, it concentrates in Strengths: 34.5% of Strengths comments are CAL versus 12.3% in Weaknesses and only 6.9% in Questions—a roughly 3$\times$ gap between Strengths and Weaknesses (Table 6). The naive mapping of section header to evaluative act fails in both directions: even Strengths sections are 65.1% descriptive ASM (most "strengths" describe what the paper does rather than commit to its

Failure mode (per model, $n = 17{,}654$ expected)	Qwen	Gemma	Nemotron
Hallucinated rows ($\notin$ expected; extras)	1	53	26
comment_id extension	1	53	13
reviewer_id fabrication	0	0	13
Schema violation (in expected set)	205	588	42
ASM `standard_relevant` = NA (only)	171	164	4
ASM `claim_relevant` = NA (only)	32	418	37
both ASM slots NA	2	2	0
CAL `grounded` = NA	0	4	1
Valid (code + slots)	17,449	17,066	17,612
Per-model valid %	98.84	96.67	99.76

Table 7: Per-model failure profiles. Models fail differently: Nemotron hallucinates reviewer identities; Qwen and Gemma dodge orthogonal Level-2 slots. Valid + schema violations sum to 17,654 per model (hallucinations are extras outside the expected set). The 3-model consensus join filters hallucinated rows; schema violations resolve through majority voting (§4.3).

value), and Weaknesses sections are 85.5% ASM (reviewers tend to *enumerate* weaknesses rather than value-judge them). The asymmetry has its own substance: reviewers articulate value commitments much more freely when complimenting (novelty, significance) than when critiquing—they describe what is wrong, not why it matters. These patterns make act annotation a non-trivial signal that section labels alone cannot recover.

These four observations motivate §6: the act layer is reliably automatable but the attribute layer requires care, and calibration is both sparse and section-asymmetric. Authors, reviewers, and area chairs all need a view of *where* value commitments live, not just *how many* comments a section contains—the basis for the audience-specific affordances.

5.2 Per-model failure analysis

Each model exhibits a *characteristic* failure mode (Table 7). All three achieve 100% coverage on the act code at the consensus stage; per-model valid% on the act code plus required-slot attributes is Qwen 98.84%, Gemma 96.67%, Nemotron 99.76%. Residual failures collapse to two axes.

Hallucinated rows—fabricated (paper, reviewer, comment_id) triples absent from the expected set—are auto-filtered at the consensus join. Nemotron distinctively fabricates reviewer IDs (13 of its 26 hallucinations).

Schema violations, where a model places NA in a required Level-2 slot, concentrate in ASM's two binary attributes with a directional pattern: Qwen dodges `standard_relevant` (171/205), Gemma dodges `claim_relevant` (418/588), Nemotron is balanced. Both ASM slots are NA in just 4 comments (Qwen 2, Gemma 2, Nemotron 0). Multi-model voting absorbs these orthogonal dodges through the gating rule described in §4.3.

6 Applications

The measured automation boundary in §5 has direct design consequences. The act layer (assessment vs. calibration) is reliably classifiable by LLMs ($\kappa \geq 0.75$); the attribute layer is not (average inter-model κ falling to 0.32 on *grounded*). This asymmetry means stakeholders can trust act-level classifications but must verify attribute-level suggestions. We outline three affordance sketches that operate within this boundary (Figure 1).

6.1 For reviewers: assessment-then-calibration workflow

The framework provides a structural self-check: does my review contain any explicit calibration, or does it consist entirely of assessments? In our corpus, 17% of reviewer–paper instances contain no calibration at all—the review describes what the paper does but never commits to why the work matters. This check operates at the act layer, where LLM classification is reliable. We release a practical reviewer guideline (Appendix D) that operationalizes this workflow, mapping assessment to a *Claim–Evidence Alignment* and *Scholarly Standard Assessment* two-step, and calibration to a *ground your judgment* injunction. Per-reviewer corpus norms from §5.1 (median 1 calibration per review; 81.4% of reviewers issue ≤ 2) provide a descriptive complement to the guideline's prescriptive recommendations.

6.2 For authors: evidence vs. argumentation

The assessment–calibration distinction maps to two response modes. Assessment comments call for *evidence*: a missing experiment can be supplied, an unsupported claim substantiated, a standard-practice gap closed. Calibration comments call for *argumentation*: the author must reframe the contribution's significance or contextualize the work's value. Treating a calibration as if it were an assessment—responding to "the novelty is insufficient" with additional ablations—addresses the wrong level. The act-level classification provides this routing; we derive five rebuttal strategies (LEVERAGE, DEFEND, CONCEDE, ACCEPT, CLARIFY) whose per-comment applicability follows from the (act, attribute) coding. The strategy taxonomy and the (act, attribute) $\rightarrow$ menu mapping are detailed in Appendix B. We frame the menu as a prospective decision-support proposal that the framework supports rather than a measured response distribution; large-scale validation against author-side rebuttal corpora, complementing recent work on author-input-driven response generation (Ruan and Gurevych, 2026), is future work.

6.3 For area chairs: discussion-phase panel view

The framework surfaces reviewer-panel structure that raw scores conceal. An area chair can see which reviewers committed value judgments and which wrote purely descriptive reviews, where calibrations cluster across the panel, and which comments the LLM panel itself disagreed on (the 11.3% with 2-of-3 consensus warrant closer reading). Given that calibration is sparse (17% zero-CAL) and section-asymmetric (35% CAL in Strengths vs. 12% in Weaknesses), surfacing where and how much explicit calibration each reviewer issued gives the AC direct access to the evaluative core of the panel. A discussion-phase dashboard rendering this information, with a worked case illustrating how a polarized four-reviewer panel ($2/4/6/8$) collapses to two traceable grounded concerns, is described in Appendix E.

6.4 Responsible use

Each affordance above relies on the act layer, where LLM consensus is reliable. Attribute-level outputs (e.g., whether a calibration is grounded, whether an assessment targets a specific claim) are advisory: they narrow the decision space but require human confirmation. The automation boundary is not a limitation to be overcome; it is the contribution. ReviewMap measures where LLM classification can be trusted and makes that boundary visible to every stakeholder.

7 Conclusion

ReviewMap operationalizes a distinction that venue reviewer guidelines have always named informally—between *describing what a paper does* and *judging why the work matters*—as a classification scheme that an open-weight LLM consensus can apply at the scale of an entire venue review cycle, with a characterized reliability boundary. The released artifact (500 papers, $17{,}654$ comments, 3-model consensus codings, frontier and human cross-validation on a development subset, and a reproducible pipeline) is, to our knowledge, the first corpus of this scale to expose the *evaluative acts* performed by reviewers at the comment level. Building on this boundary, we sketch three role-specific affordances—a reviewer guideline, an author rebuttal scaffold, and an AC dashboard—that operate strictly within the automatable layer. Beyond the resource itself, we report two structural facts about ICLR reviewing that emerge from this view: explicit value commitments are sparse (17% of reviewer–paper instances contain no calibration; median 1 CAL per review) and section-asymmetric ($3\times$ more calibration in Strengths than in Weaknesses). We argue that *measuring the automation boundary*—where LLM classification can be trusted and where it cannot—defines a more honest mission for NLP in peer review than full automation, complementing the rigor call of Baumann et al. (2026).

Limitations

ReviewMap classifies evaluative acts; it does not score reviews, rank reviewers, or predict accept/reject. The 17% zero-CAL finding (§5) is a measurement of writing practice, not a critique of reviewers. Score prediction is out of scope: ICLR 2026 experienced a venue-level data event during the cycle that left only post-discussion final scores accessible in the public API, so probing predictive signal at a venue with pre/post score visibility—as done in concurrent ICLR rebuttal–score analyses (Kargaran et al., 2025)—is future work. Reliability is bounded along the objectivity gradient: the act layer is reliable enough to automate (inter-

model $\kappa \approx 0.75$–0.77, frontier-vs-local $\kappa = 0.84$, 88.4% unanimous consensus), but the Level-2 attribute layer is model-bound (κ falls to Fair at *grounded*). Applications dependent on attribute-layer judgments include a human-in-the-loop component (§6.4) and should not be deployed without it.

The dev-set single ML-expert annotator (mirroring the area-chair reading condition) reaches Substantial agreement with both consensus panels ($\kappa = 0.65$ frontier, $\kappa = 0.62$ local), while inter-LLM agreement is higher ($\kappa = 0.84$). The $\Delta\kappa \approx 0.2$ gap is consistent with the multi-annotator IRR ceiling in peer-review NLP (Kennard et al., 2022, $\kappa = 0.447$–0.605) and with score-level dispersion in conference review more broadly (Beygelzimer et al., 2023; Cortes and Lawrence, 2021); we report it as a feature of subjective-annotation IRR rather than a shared-bias signal. We characterize ICLR 2026 only; ARR/EMNLP-derived corpora require the Yes–Yes–Yes donation protocol (Dycke et al., 2023), and a multi-venue extension—building on recent cross-venue review corpora such as Re2 (Zhang et al., 2025)—is the natural next step.

The reviewer-, author-, and area-chair-side applications in §6 are framework-derived sketches with a worked AC dashboard case (Appendix E); a deployment study is future work. Rule-based segmentation occasionally produces fragmentary atomic units, isolated by FLAG ($\sim$2.3% of dev-set comments). The schema has a known design tension between `standard_relevant` (ASM P2) and `grounded` (CAL) — targeting versus justification. Cross-model shared training-data bias cannot be ruled out; cross-family selection (Qwen, Gemma, Nemotron) partially mitigates this. Robustness to guideline variation is not tested.

Ethical considerations

All data is drawn from ICLR 2026 submissions on OpenReview, publicly available under OpenReview's research-use terms; no additional IRB review or per-author consent was sought, consistent with prior peer-review NLP corpora that release similarly sourced ICLR data (Kennard et al., 2022; D'Arcy et al., 2024; Dycke et al., 2023). The released artifact retains paper identifiers and the venue's public pseudonymous reviewer identifiers (e.g., `Reviewer_NydX`) unchanged, matching the DISAPERE / ARIES convention; we do not perform additional pseudonymization and do not attempt to de-anonymize reviewers. Case studies in Appendix E and Appendix B quote short illustrative fragments and reduce per-paper reviewer labels to short forms (R1, R2, ...) for readability; full review and rebuttal threads remain accessible at the OpenReview source URLs.

Risks and intended-use scope. ReviewMap is intentionally a classification framework, not a generation system: the released code does not synthesize review or rebuttal text. We share the position of Dycke et al. (2023) and (Baumann et al., 2026) that automatic review generation is a category of NLP application where benefits are unclear and risks to academic quality assurance are real. The reviewer-side, author-side, and area-chair-side applications in §6 are diagnostic decision-support layers; they do not write text on behalf of users. The released corpus is not intended for reviewer profiling, for automated decision-making in peer-review venues, or for training general-purpose language models.

References

Anthropic. 2026. Claude Opus 4.6 system card. https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf. Accessed: May 2026.

Joachim Baumann, Jiaxin Pei, Sanmi Koyejo, and Dirk Hovy. 2026. Stop automating peer review without rigorous evaluation. *arXiv preprint arXiv:2605.03202*.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Alina Beygelzimer, Yann N Dauphin, Percy Liang, and Jennifer Wortman Vaughan. 2023. Has the machine learning review process become more arbitrary as the field has grown? the neurips 2021 consistency experiment. *arXiv preprint arXiv:2306.03262*.

Prabhat Kumar Bharti, Meith Navlakha, Mayank Agarwal, and Asif Ekbal. 2024. Politepeer: Does peer review hurt? a dataset to gauge politeness intensity in the peer reviews. *Language Resources and Evaluation*, 58(4):1291–1313.

Aakshita Chandiramani, Aaron Blakeman, Abdullahi Olaoye, Abhibha Gupta, Abhilash Somasamudramath, Abhinav Khattar, Adeola Adesoba, Adi Renduchintala, Adil Asif, Aditya Agrawal, and 1 others. 2026. Nemotron 3 super: Open, efficient mixture-of-experts hybrid mamba-transformer model for agentic reasoning. *arXiv preprint arXiv:2604.12374*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scale. *Educ Psychol Meas*, 20:37–46.

Corinna Cortes and Neil D Lawrence. 2021. Inconsistency in conference peer review: Revisiting the 2014 neurips experiment. *arXiv preprint arXiv:2109.09774*.

Delphine S Courvoisier, Diana Buitrago-Garcia, Clément P Buclin, Nils Bürgisser, Michele Iudici, and Denis Mongin. 2026. Beyond human gold standards: A multimodel framework for automated abstract classification and information extraction. *Research Synthesis Methods*, 17(2):365–377.

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2024. ARIES: A corpus of scientific paper edits made in response to peer reviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6985–7001, Bangkok, Thailand. Association for Computational Linguistics.

Google DeepMind. 2026. Gemma 4: Open multimodal models. `https://ai.google.dev/gemma/docs/core/model_card_4`. Accessed: May 2026.

Nils Dycke, Ilia Kuznetsov, and Iryna Gurevych. 2023. NLPeer: A unified resource for the computational study of peer review. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5049–5073, Toronto, Canada. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Google DeepMind. 2025. Gemini 3 Pro model card. `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf`. Accessed: May 2026.

Yanzhu Guo, Guokan Shang, Virgile Rennard, Michalis Vazirgiannis, and Chloé Clavel. 2023. Automatic analysis of substantiation in scientific peer reviews. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10198–10216, Singapore. Association for Computational Linguistics.

Xinyu Hua, Mitko Nikolov, Nikhil Badugu, and Lu Wang. 2019. Argument mining for understanding peer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2131–2137, Minneapolis, Minnesota. Association for Computational Linguistics.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. AgentReview: Exploring peer review dynamics with LLM agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1208–1226, Miami, Florida, USA. Association for Computational Linguistics.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661, New Orleans, Louisiana. Association for Computational Linguistics.

Amir Hossein Kargaran, Nafiseh Nikeghbal, Jing Yang, and Nedjma Ousidhoum. 2025. Insights from the iclr peer review and rebuttal process. *arXiv preprint arXiv:2511.15462*.

Neha Nayak Kennard, Tim O'Gorman, Rajarshi Das, Akshay Sharma, Chhandak Bagchi, Matthew Clinton, Pranay Kumar Yelugam, Hamed Zamani, and Andrew McCallum. 2022. DISAPERE: A dataset for discourse structure in peer review discussions. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1234–1249, Seattle, United States. Association for Computational Linguistics.

Jaeho Kim, Yunseok Lee, and Seulki Lee. 2025. Position: The ai conference peer review crisis demands author feedback and reviewer rewards. *arXiv preprint arXiv:2505.04966*.

Ilia Kuznetsov, Osama Mohammed Afzal, Koen Dercksen, Nils Dycke, Alexander Goldberg, Tom Hope, Dirk Hovy, Jonathan K Kummerfeld, Anne Lauscher, Kevin Leyton-Brown, and 1 others. 2024. What can natural language processing do for peer review? *arXiv preprint arXiv:2405.06563*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel Mcfarland, and James Y. Zou. 2024. Monitoring AI-modified content at scale: A case study on the impact of ChatGPT on AI conference peer reviews. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29575–29620. PMLR.

Sheng Lu, Ilia Kuznetsov, and Iryna Gurevych. 2025. Identifying aspects in peer reviews. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6145–6167, Suzhou, China. Association for Computational Linguistics.

OpenAI. 2026. GPT-5.5 system card. https://openai.com/index/gpt-5-5-system-card/. Accessed: May 2026.

Sukannya Purkayastha, Zhuang Li, Anne Lauscher, Lizhen Qu, and Iryna Gurevych. 2025. LazyReview: A dataset for uncovering lazy thinking in NLP peer reviews. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3280–3308, Vienna, Austria. Association for Computational Linguistics.

Qian Ruan and Iryna Gurevych. 2026. Author-in-the-loop response generation and evaluation: Integrating author expertise and intent in responses to peer review. *arXiv preprint arXiv:2602.11173*.

David Soergel, Adam Saunders, and Andrew McCallum. 2013. Open scholarship and peer review: A time for experimentation. https://openreview.net/forum?id=xf0zSBd2iufMg. ICML 2013 Peer Review Workshop.

Qwen Team. 2026. Qwen3.5-omni technical report.

Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. 2026. A large-scale randomized study of large language model feedback in peer review. *Nature Machine Intelligence*, pages 1–11.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Sihong Wu, Yiling Ma, Yilun Zhao, Tiansheng Hu, Owen Jiang, Manasi Patwardhan, and Arman Cohan. 2026. Rbtact: Rebuttal as supervision for actionable review feedback generation. *arXiv preprint arXiv:2603.09723*.

Weizhe Yuan, Pengfei Liu, and Graham Neubig. 2022. Can we automate scientific reviewing? *Journal of Artificial Intelligence Research*, 75:171–212.

Daoze Zhang, Zhijian Bao, Sihang Du, Zhiyi Zhao, Kuangling Zhang, Dezheng Bao, and Yang Yang. 2025. Re2: A consistency-ensured dataset for full-stage peer review and multi-turn rebuttal discussions. *arXiv preprint arXiv:2505.07920*.

A Reproducibility

Implementation. All three local models are served with vLLM (BF16). We use a single decoding configuration across all self-consistency runs: temperature 0.6, vLLM-default `top_p`/`top_k`, and per-model maximum output tokens set to each model's stable output length on our prompts — Qwen 16K, Nemotron 20K, Gemma 4K. No hyperparameter search was performed. Serving scripts and the exact decoding configuration are released with the software bundle.

Hardware and compute budget. Annotation ran on two hardware configurations: a workstation with dual NVIDIA RTX Pro 6000 Blackwell (96 GB each) hosting Qwen and Gemma on a single GPU each, and a server with dual NVIDIA B200 (192 GB each) hosting Nemotron at tensor-parallel TP=2. Table 8 reports the per-model compute budget in GPU-hours (the standard cluster-accounting convention, accounting for Nemotron's TP=2). The full 3-model $\times$ 3-run $\times$ 500-paper main-set annotation consumed $\approx$352 GPU-hours total, averaging 42.4 GPU-minutes per paper across all nine model passes. Qwen timings are directly measured from per-run logs; Gemma and Nemotron figures are estimated from truncation-retry timing samples on the production max-token settings (logs of the full main runs were rotated during the truncation fix).

Seeds and determinism. Stage 1 corpus sampling (§4.1) is deterministic at `SEED=42`; re-running the sampler with the same seed reproduces the same 500-paper sample. Stage 2 LLM annotation at temperature 0.6 is non-deterministic, so re-runs against the same models produce *near-identical* but not bit-identical codings. We therefore release the exact codings used in the paper (per-model voted files and the 3-model consensus) and recommend agreement against them as the appropriate reproducibility metric; we observe $> 95\%$ act-code agreement on independent re-runs of single models against the released voted files.

B Author-Side Strategy Scaffold: Detail

This appendix expands §6.2. We provide (i) full strategy definitions, (ii) the coding→menu mapping, (iii) venue-template adaptation, and (iv) convergence with RbtAct's independently derived rebuttal-action taxonomy. The scaffold is a prospective decision-support proposal that the framework supports; large-scale validation against author-side rebuttal corpora is left to future work.

B.1 Five-strategy definitions

We define five candidate response strategies for the author's write-time decision space; their cardinality and structural axes are independently corroborated by RbtAct (see convergence discussion below):

- LEVERAGE — quotes the comment as third-party support in the opening of the rebuttal.
- DEFEND — pushes back with citation, data, or precedent.
- CONCEDE — accepts a substantive critique and offers a concrete remedy.
- ACCEPT — low-effort revision commit (typo, formatting) with no epistemic concession.
- CLARIFY — answers directly with paper content or asks the reviewer for clarification.

ACCEPT and CONCEDE both involve agreement, distinguished by depth: ACCEPT is purely surface-level (no model/claim revision), while CONCEDE signals a substantive update to the paper or its scope.

B.2 Coding → strategy menu mapping

The (code, attribute) coding narrows the menu to 2–4 applicable strategies per comment (Table 9). The author picks one.

Approximately 5–6% of comments fall to the author's direct judgment (FLAG 2.2% + no-majority codings 0.3% + Level-2 schema violations $\sim$ 3% in the most-affected model). The remaining 94–95% receive a menu. Within scaffolded comments, the LLM panel's unanimity modulates trust: unanimous codings (88.4%) license the menu directly; 2-of-3 codings (11.3%) apply on majority basis but the author confirms by reading the comment text.

B.3 Venue-template adaptation

Section-template adaptation is a venue-specific prefilter on top of the (code→menu) mapping: ICLR's Strengths/Weaknesses/Questions template makes the menu choice fast (Strengths → LEVERAGE; Weaknesses → DEFEND/CONCEDE/ACCEPT; Questions → CLARIFY); ARR's heterogeneous Comments/Suggestions/Typos sections require the coding for routing; CVPR-style no-section rebuttals work entirely on the coding. The (coding

Model (BF16)	Hardware	GPUs	Per single pass (GPU-min)	Per paper (3 runs) (GPU-min)	Full pipeline (GPU-h)
Qwen 3.5–27B	RTX Pro 6000 (96 GB)	1	8.5	25.5	**212.3**
Gemma 4–31B	RTX Pro 6000 (96 GB)	1	0.7	2.0	**16.3**
Nemotron 3-Super 120B	NVIDIA B200 (192 GB)	2	5.0	14.9	**124.0**
Total (all 3 models)	two hardware configurations	—	**14.2**	**42.4**	≈ 352

Table 8: Compute budget for the main-set annotation, in GPU-hours throughout. *Per single pass* = one paper through one model in one SC run; *per paper (3 runs)* = one paper through all three SC runs of one model; *full pipeline* = the entire 3-model $\times$ 3-run $\times$ 500-paper annotation that produced the released artifact. Nemotron's TP=2 contributes two GPU-hours per elapsed hour.

Coding	Strategy menu
ASM, P1=yes	LEVERAGE · DEFEND · CONCEDE · CLARIFY
ASM, P2=yes	LEVERAGE · DEFEND · CONCEDE · CLARIFY · ACCEPT
CAL	LEVERAGE · DEFEND · CONCEDE
FLAG	(no scaffold — author reads directly)

Table 9: Coding $\rightarrow$ strategy menu mapping. Each (act, attribute) coding selects 2–4 applicable strategies; the author picks one from the menu.

$\rightarrow$ menu) mapping is the venue-invariant core. The author still writes the rebuttal; the scaffold makes the strategy budget visible per paper ("7 LEVERAGE candidates, 11 DEFEND/CONCEDE pending, 18 CLARIFY pending").

B.4 Convergence with RbtAct as external validation

Our prospective 5-strategy decision menu (LEVERAGE/DEFEND/CONCEDE/ACCEPT/CLARIFY) converges on a similar five-fold cardinality to RbtAct's (Wu et al., 2026) independently derived rebuttal action taxonomy (CRP / SRP / VCR / DWC / DRF), with axis-by-axis structural correspondence rather than a strict one-to-one mapping. DEFEND aligns with RbtAct's *Defend Without Change* (DWC) and partially with *Deflect/Reframe* (DRF); CONCEDE aligns with the depth-graded *Concrete Revision Performed* (CRP) $\rightarrow$ *Specific Revision Plan* (SRP) $\rightarrow$ *Vague Commitment to Revise* (VCR) triple, which RbtAct splits by realized impact magnitude. Two principled differences are themselves informative. First, RbtAct's taxonomy labels only reactions to criticism, so LEVERAGE (response to positive strengths) has no RbtAct counterpart — this is the complement that a prospective decision menu, written before authoring, must cover. Second, RbtAct splits concession by impact across three categories, whereas ReviewMap collapses to one CONCEDE plus a separate ACCEPT (low-effort revision commit) on the rationale that the author's decision space at write-time is closer to binary (concede / not concede), while the realized impact gradient is a post-hoc observable. The two systems were developed for different goals — RbtAct labels rebuttals to supervise feedback-generation models for reviewers; ReviewMap surfaces a decision menu before the author writes — yet converge on the same axes of action. Independent convergence from different methodological starting points provides external evidence that these categories carve the author-response space along its natural joints, not arbitrary divisions.

C Annotation Guideline for ReviewMap

This appendix reproduces the full text of the annotation guideline used to produce the released codings (§4.2, §5). The same text was provided in the system prompt to each LLM annotators. The guideline below uses the term *evaluative function*; the body of this paper refers to the same concept as *evaluative acts*.

Annotation Guideline (released artifact)

C.1 Purpose

This guideline provides instructions for classifying peer-review comments from ML/CS venues by the *evaluative function* each comment performs: **Assessment (ASM)** or **Calibration (CAL)**.

- **ASM** evaluates *what the paper does or how it does it* — whether claims are supported and standards are met.
- **CAL** evaluates *why the work matters* — the contribution's novelty, significance,

importance, or venue fit.

C.2 Output format

Each comment receives the following fields:

Field	Values	When
`code`	ASM / CAL	Always
`claim_relevant`	yes / no	ASM only
`standard_relevant`	yes / no	ASM only
`grounded`	yes / no	CAL only
`note`	free text	When deliberating between codes, or when recording grounding evidence

Coding unit. Each individual comment as delimited by the reviewer under Strengths (S1, S2, . . .), Weaknesses (W1, W2, . . .), or Questions (Q1, Q2, . . .). Code only official reviews.

C.3 Identifying claims (required before coding)

Before coding any comments, read the paper's **abstract**, **introduction**, and **conclusion** and list the main claims. A claim is any assertion the paper explicitly makes — empirical results, scope, design properties, or performance characteristics.
Examples of claims:

- "achieves state-of-the-art performance on X"
- "our method is model-agnostic and can be applied to any architecture"

This claim list is your reference for determining `claim_relevant` below.

C.4 Category definitions

C.4.1 Assessment (ASM)

A comment is ASM when it evaluates *what the paper does or how it does it* — whether claims are supported, methods are sound, standards are met, or presentation is adequate. ASM examines the paper's content against its own assertions or against accepted scholarly practices.
To recognize ASM, ask two questions:

- Does this comment evaluate whether a *specific claim* is supported by evidence? (paper vs. its own assertions)
- Does this comment evaluate whether the paper meets a *field-wide standard*? (paper vs. accepted practices)

If either answer is yes, the comment is ASM. *ASM encompasses both positive and negative assessments:*

- "The analysis clearly demonstrates the contribution of each proposed component." — evaluates what the paper did (claim support).
- "The paper claims real-time performance but the reported latency excludes preprocessing time." — evaluates what the paper claimed vs. showed.
- "The paper is well-written and easy to follow." — evaluates how the paper presents its work.
- "The experimental setup is not described in sufficient detail for reproduction." — evaluates what is missing (standard).

C.4.2 Calibration (CAL)

A comment is CAL when it evaluates *why the work matters* — positioning the contribution in the field by judging its novelty, significance, importance, or venue fit.
CAL encompasses both positive and negative calibrations:

- "This is the first computational work to link X and Y." — positions the work as novel; the point is why it matters, not what was done.
- "The novelty is limited — similar approaches have been widely explored." — judges why the work does not matter (novelty).
- "This will change how the field thinks about X." — judges why the work matters (significance).
- "The contribution is incremental." — judges why the work does not matter (significance).

C.5 Distinguishing ASM vs. CAL

When a comment touches both assessment and calibration, code the *dominant evaluative function*. To determine dominance, ask: is the comment ultimately about *what the paper does*, or about *why the work matters*?
Use the `note` field to record any secondary dimension you considered. Interrogative form does not determine the label — questions should be coded by the evaluative function they perform.

Comment	Code	Reasoning
"This is an invaluable contribution with careful detail and rigor"	CAL	Mixed: ultimately about why the work matters (invaluable), not about the detail itself
"The results are impressive but the comparison with prior work is unfair"	ASM	Mixed: ultimately about what the paper does (unfair comparison), not about impressiveness

C.6 ASM attributes

For each comment coded ASM, record two attributes:

`claim_relevant`. Does the comment evaluate the support for a *specific claim* listed above? (yes / no). This checks the paper against *its own assertions* — did the paper do what it said?

`standard_relevant`. Does the comment evaluate whether the paper meets a *field-wide standard* that applies regardless of what the paper claims? (yes / no). This checks the paper against *accepted scholarly practices* — did the paper do what is expected?
Both can be yes simultaneously. At least one must be yes; if neither applies, reconsider whether the comment is CAL.

C.7 CAL attribute

For each comment coded CAL, record:

`grounded`. Does the review text contain *traceable evidence* supporting the value judgment? (yes / no). If yes, briefly note the evidence in the `note` field.
Traceable evidence includes the reviewer's own assessment of the paper, a specific reference to the paper's content, or a comparison with other work. Identifying the feature being evaluated does not by itself count as evidence.

- Grounded: "The improvement is marginal — only X% over the compared method." (evidence: inline, X% figure)
- Ungrounded: "This is a strong contribution to the field."

C.8 Uncertain cases (FLAG)

If you cannot determine whether a comment is ASM or CAL, output FLAG as the code.

D ReviewMap as Reviewer Guideline

This appendix presents the structure and core content of the *Practical Reviewer Guideline* (§6.1).

Practical Reviewer Guideline (released artifact)

D.1 Before you write: reading the paper

A review is a response to what the paper actually says, not to what you wish it said. Before writing anything, identify the following:

- *What does the paper claim?* Read abstract, introduction, and conclusion carefully. List the specific claims the authors make.
- *What evidence does the paper provide?* For each claim, identify the corresponding evidence: experiments, proofs, ablations, visualizations, or theoretical arguments.
- *What is the closest prior work?* Identify the most similar existing work. Understanding how the paper positions itself relative to this work is often the key to evaluating its contribution.

Take notes on these three points before you start writing. They form the backbone of the review.

D.2 The review process: a step-by-step workflow

Reviewing is a structured process. The workflow below assumes the reviewer's venue permits LLM assistance for preparation and polish but prohibits delegating judgment to LLMs (e.g., ICML 2026 Policy B; NeurIPS,

ICLR, ACL); *the venue's policy always takes precedence* (some venues prohibit all LLM use, e.g., ICML 2026 Policy A).

Phase 1 — Preparation (AI assistance generally permitted).

1. Read the abstract and decide whether to accept the review. Decline if outside your area of competence.
2. Build your reference landscape. List cited related works; use AI to search for additional recent works.
3. Familiarize yourself with key references using AI assistance.
4. Prepare comparison points: what each related work does differently, where claims overlap, how the proposed method should be positioned.

Phase 2 — Reading the paper (no AI assistance).

5. Read the paper yourself, with your prepared comparison points. Most venues prohibit LLMs during this phase, and reading the paper through an AI's summary risks losing the independent judgment that peer review exists to provide.

Phase 3 — Post-reading verification (AI for targeted queries).

6. Check for additional missing references. Your understanding after reading the paper allows precise queries.

Phase 4 — Writing and finalizing.

7. Write the review in your own words.
8. Optionally use AI for final polish (grammar, clarity, score-consistency, unintentional tone issues). This is verification, not generation.
9. Submit.

D.3 Structure of a review

A well-structured review serves authors, the area chair, and other reviewers:

- **Summary (1 paragraph).** What the paper does in your own words: problem, approach, key results.
- **Strengths (2–5 points).** Specific strengths tied to concrete aspects. *Bad:* "The experiments are thorough." *Good:* "The ablation in Table 3 isolates the contribution of each component across three datasets."
- **Weaknesses, separated as major (affecting validity of core claims) vs. minor (presentation, secondary).** Be specific about why each issue matters.
- **Questions to authors.** Frame as genuine questions, not rhetorical attacks.
- **Suggestions for improvement** (optional but recommended).
- **Minor issues** (typos, formatting; keep brief).

D.4 Two core principles for evaluating a paper

A review answers two questions about a paper. The review proceeds in two stages:

1. **Assessment** (this section): assess what the paper did and how it did it, using Principles 1 and 2.
2. **Calibration** (§D.5): based on your assessments, make subjective value judgments about novelty, significance, and overall recommendation.

The ordering is not arbitrary: novelty cannot be judged without first identifying what the paper claims (Principle 1); significance cannot be judged without first confirming that the claims are supported. P1/P2 assessments are *verifiable* (a different reviewer could confirm or refute them); calibration judgments are *inherently subjective* (two competent reviewers may legitimately disagree).

D.4.1 Principle 1: Claim–Evidence Alignment

"Does this paper sufficiently support what it claims?"

The authors set the bar by making specific claims; the reviewer checks whether the evidence meets that bar. Steps: (1) identify each claim (empirical, scope, design, or compliance assertions); (2) find the corresponding evidence; (3) assess the gap. Common patterns range from well-supported (acknowledge as strength), through overclaimed (point out the specific mismatch) and incomplete evidence (request the specific missing experiment), to no evidence (flag as major concern) and evidence contradicting the claim (point out precisely with references). *Scale your concern to the size of the gap.*

D.4.2 Principle 2: Scholarly Standard Assessment

"Does this paper meet the standards that apply regardless of what it claims?"
Some qualities are expected of every academic paper, whether or not the authors explicitly address them. Key areas: reproducibility; ethical considerations; fair treatment of prior work (citation accuracy, comparison fairness); honest acknowledgment of limitations; writing quality. A paper might pass Principle 1 perfectly yet still fall short on Principle 2.

D.4.3 How the two principles work together

In practice, most comments fall under one principle or the other; some fall under both. *Examples:* "Claims state-of-the-art performance but does not compare with Method X" → Principle 1 (claim without evidence) and Principle 2 (unfair treatment of prior work); "No code or hyperparameters provided" → Principle 2 (scholarly standard); "Claims robustness across diverse domains but evaluates only on one narrow dataset" → Principle 1. You do not need to label each comment; the principles are a thinking tool, not a formatting requirement.

D.5 Calibrating your judgment

After completing P1/P2 assessment, make subjective value judgments. *Ground each judgment* by citing the P1 or P2 assessment that supports it.

Novelty. *"Is this contribution new relative to existing knowledge?" Good:* "This is, to my knowledge, the first work to combine X with Y." *Good (with citation):* "Method Z [Ref 12] already addresses the same problem with a similar approach, which limits the novelty of this submission." *Bad:* "The novelty is limited." (Ungrounded.)

Significance. *"Does this contribution meaningfully advance the field or community?"* A paper can be novel but insignificant (first solution to a trivial problem) or significant but not novel (large-scale validation of a known technique). Significance includes venue-relative impact.

Confidence. Be honest about your confidence level. A wrong assessment with high confidence is worse than a correct one with moderate confidence.

Severity of language. Match the strength of your language to the strength of your evidence. A mathematical proof of an error supports direct statements; a suspicion supports a question or request for clarification; a matter of taste should be flagged as such.

Overall recommendation. Should *follow from* P1/P2 assessment + novelty/significance judgments, not precede them. If your recommendation conflicts with your stated assessments (no major weaknesses but recommend rejection; identified fatal flaws but recommend accept), revisit your assessment for consistency.

D.6 Common mistakes in reviewing

- **Reviewing the paper you wanted, not the paper you got.** Evaluate a task-definition paper as a task-definition paper, not as a methods paper.
- **Penalizing novelty and simplicity.** Novel/interdisciplinary work is easy to criticize but may be important long-term. "Too simple" is often retrospective: ask whether the specific combination has been tried before.
- **Using common sentiments without evidence.** "This work is not new" — cite

the prior work. "This work is not enough" — specify what additional contribution is needed and why. "There is no [X] experiment" — tie to a specific claim, not a generic template.

- **Listing problems without acknowledging strengths.** Every published paper has weaknesses, and most rejected papers have strengths.
- **Confusing "I don't know this" with "this is wrong".** Look it up or note that you cannot evaluate this aspect.
- **Demanding experiments irrelevant to the paper's claims.** "More experiments" is not useful; "an ablation comparing X vs. Y would clarify whether the improvement comes from component A or B" is.
- **Using dismissive language.** The content of criticism can be direct; only the framing changes. "This is trivial" → "The technical contribution is primarily in system design"; "The authors failed to" → "The paper does not currently include".

D.7 Pre-submission checklist

- Comprehension: does my summary accurately reflect what the paper does?
- Balance: have I identified both strengths and weaknesses?
- Specificity: is every weakness tied to a specific section, table, equation, or claim?
- Proportionality: are major concerns truly major; are minor issues kept minor?
- Actionability: would the authors know how to improve their paper from my review?
- Grounded value judgments: when I state a novelty or significance judgment, have I cited the P1 or P2 assessment that supports it?
- Assessment before judgment: did I complete P1/P2 before forming my overall recommendation?
- Tone: would I be comfortable receiving this review for my own paper?
- Consistency: does my rating match the severity of the issues I raised?
- Confidence: does my confidence score honestly reflect my expertise on this topic?

E Area-Chair Dashboard Mockup example

This appendix presents a mockup example of the Area-Chair Dashboard (Fig. 4), drawn from one paper in our main corpus.

```
+-------------------------------------------------------------------------+
|  ReviewMap -- Area-Chair Dashboard                                      |
|  Paper:  v2oYZJ7Exo     Pre-AC:  borderline (panel scores 2 / 4 / 6 / 8)  |
+-------------------------------------------------------------------------+
|                                                                         |
|  Per-reviewer profile                                                   |
|  +----------+-------+------+-------+-----+-----+------+-------+--------+  |
|  | Reviewer | Score | Conf | Sound | ASM | CAL | FLAG | gnNEG |  gnPOS |  |
|  +----------+-------+------+-------+-----+-----+------+-------+--------+  |
|  | G88g     |   8   |   5  |   3   |  8  |  3  |   0  |   0   |    0   |  |
|  | 4Vi5     |   6   |   4  |   3   |  4  |  1  |   0  |   0   |    0   |  |
|  | DEPu     |   4   |   4  |   2   | 12  |  0  |   0  |   0   |    0   |  |
|  | LnWH     |   2   |   4  |   2   |  8  |  4  |   0  |   2   |    0   |  |
|  +----------+-------+------+-------+-----+-----+------+-------+--------+  |
|                                                                         |
|  !  Asymmetric anchoring                                                |
|     - Grounded evaluative content concentrated in LnWH (score 2).       |
|     - The score-8 reviewer issued 3 CALs, all ungrounded.               |
|                                                                         |
+-------------------------------------------------------------------------+
|                                                                         |
|  Grounded negative concerns -- rebuttal trace                           |
|                                                                         |
|  > LnWH/W1   "Trivial target properties (vdW-sphere derivable)"         |
|     +- Author R1:  "Scope is adsorption capacity / separation, not      |
|                     geometric quantities; Table 1 focuses on non-trivial"  |
|        Strategy: CLARIFY     Resolution: scope misread  ->  resolved    |
|                                                                         |
|  > LnWH/W7   "Weak empirical gains, unwarranted complexity"             |
|     +- Author R5:  "Monotonic > 30% MAE reduction (Fig 7A); 30% added   |
|                     parameter complexity"                               |
|        Strategy: DEFEND      Resolution:  partial                       |
|                                                                         |
|  Concern coverage:  2 / 2 grounded NEG concerns have traced rebuttal    |
|                     segments                                            |
|                                                                         |
+-------------------------------------------------------------------------+
|                                                                         |
|  Off-task log                                                           |
|   i  1 third-party comment (OpenReviewers account, authors disavowed)   |
|                                                                         |
+-------------------------------------------------------------------------+

  Columns:
    ASM / CAL / FLAG  -- per-reviewer Level-1 code counts
    gnNEG             -- grounded negative CALs   (most diagnostic; S5.3, S6.3)
    gnPOS             -- grounded positive CALs
    Strategy badges   -- one of {LEVERAGE, DEFEND, CONCEDE, ACCEPT, CLARIFY}
    Resolution        -- manual cell filled by the AC; everything else
                         is computed from the consensus codings and
                         segmented rebuttal text.
```

Figure 4: AC dashboard mockup on paper id `v2oYZJ7Exo` (§6.3). Discussion-phase view of a polarized 2 / 4 / 6 / 8 panel. The per-reviewer profile (top) exposes that all grounded negative evaluative content is concentrated in the score-2 reviewer, while the score-8 reviewer's enthusiasm has no anchored evidence; the rebuttal-trace panel (middle) maps each grounded concern to the matched author response and records the strategy label (§6.2) plus a manual resolution status; the off-task log (bottom) surfaces a third-party intervention that occurred in this paper's discussion thread. Everything except the manual resolution status is computed from the consensus codings and segmented rebuttal text; the dashboard surfaces structure for the AC and does not recommend an outcome.