PEERREVEAL: LEARNING TO GENERALIZE IN AI PEER REVIEW DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Can we distinguish fully AI-generated reviews from those using AI only to rewrite human judgement? Furthermore, can we do so reliably when reviews are generated from diverse AI models and workflows across scientific domains? To this end, we introduce PEERREVEAL, a new detector that grounds its decisions in learned relationships between a review's claims and reference claims drawn from AI-generated reviews of the same paper, complemented by review-level textual signals. We evaluate PEERREVEAL on PeerShift, a new, challenging test set of 5,430 reviews spanning multiple domains, including machine learning, life sciences, and social sciences, combining variation in disciplinary review conventions with diverse AI review-production mechanisms. With regular training, PEERREVEAL achieves 70.52% pooled Macro-F1 on this benchmark, improving over the strongest baseline by 11.48 points. We develop an effective and practical adversarial training framework that further improves the detector's generalization using a small local generator optimized to produce challenging reviews, raising pooled Macro-F1 from 70.52% to 82.40%. These results show that robust AI-review detection benefits from richer review-level evidence and adversarial training on more challenging generation processes.

1 INTRODUCTION

Peer review is central to scientific progress. As large language models become increasingly capable of reading papers and producing detailed evaluations, LLM assistance in scientific reviewing now ranges from polishing language grounded in human judgement to generating an assessment end to end (Liang et al., 2024; Zhou et al., 2026). This range of practices makes review provenance increasingly difficult to infer: similar-looking reviews may reflect human judgements polished or expanded by an LLM, or critiques and judgements generated by the model itself.

Existing scientific-review detectors already exploit manuscript context and review-specific signals (Kumar et al., 2024), including comparisons with AI-generated reviews of the same paper as references, but two challenges remain. First, reference-based approaches usually compare reviews at the whole-review level or summarize claim-level comparisons through predefined features, rather than learning directly from interactions between individual target and reference claims (Yu et al., 2026; Duarte et al., 2026). Second, existing evaluations are typically built around a small, fixed set of models and prompting procedures selected in advance, even though real-world review production may involve models and workflows that cannot be anticipated when a benchmark is constructed.

We address the first challenge with PEERREVEAL (Figure 1), a blockwise-masked Transformer (Vaswani et al., 2017) that, through directed attention, learns directly from interactions between individual claims in the target review and in same-paper AI reference reviews. Using the ML-only regular training corpus of SEM-DETECT (Duarte et al., 2026), PEERREVEAL improves Macro-F1 by 11.83 points on the in-distribution evaluation, 21.55 points with unseen writer models, and 18.84 points with unseen writer models and prompts, demonstrating its architectural benefit.

Although these results are strong under controlled writer-model and prompt shifts, they do not establish whether the detector remains reliable when review production departs from the conditions anticipated during benchmark construction. For that reason, we construct PeerShift, a broader, deliberately challenging test set, using an automatic search method that discovers diverse ways of generating AI reviews and rewriting human reviews while maintaining review quality. The resulting

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

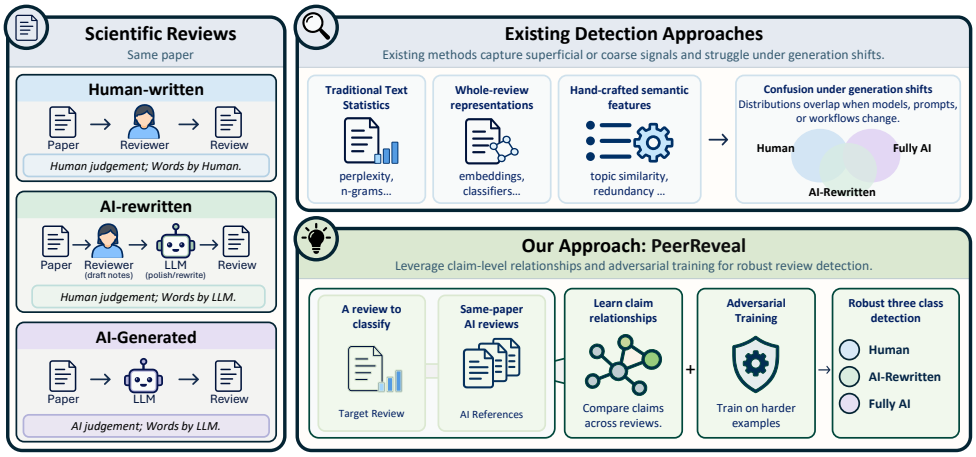


Figure 1: PEERREVEAL learns claim relationships and uses adversarial training to distinguish human-written, AI-rewritten, and AI-generated reviews across models, workflows, and domains.

evaluation spans different writer models and comprises a balanced set of 5,430 reviews across machine learning, life sciences, and social sciences. On its ML subset (PeerShift-ML), PEERREVEAL trained only on the original ML data achieves 48.14% Macro-F1, outperforming SEM-DETECT by 13.74 points, but leaving a substantial generalization gap.

This remaining gap suggests that detector architecture alone is insufficient when training is confined to a fixed distribution. We therefore develop an adversarial generator–detector training framework in which a small local generator is optimized to produce difficult, high-quality reviews that expose weaknesses in the detector. On PeerShift, adversarial training improves pooled Macro-F1 from 70.52% with regular training to 82.40%, demonstrating an effective and practical approach to improving detector generalization with a small local generator.

Our contributions are:

1. **A stronger peer-review authorship detector.** PEERREVEAL directly learns interactions between individual claims in target and same-paper AI reference reviews, substantially improving detection under both in-distribution and controlled distribution-shift settings.
2. **Systematic construction of challenging peer-review evaluations.** We automatically search for diverse review-production mechanisms under quality constraints, yielding PeerShift, a cross-domain test set probing generalization beyond model and prompt shifts.
3. **A practical adversarial training framework for improving generalization.** Our framework adversarially optimizes a small local generator to produce difficult training examples, providing an effective source of additional supervision and substantially improving detection under previously unseen review-production conditions.

2 RELATED WORK

Detecting AI-generated text has prompted research on watermarking, statistical detection, and supervised classification (Wu et al., 2025). We first review general-purpose detectors and methods for scientific peer review, and then discuss generalization and adversarial training.

General-purpose AI-text detection. Watermarking embeds detectable signals during text generation, but requires control over the generation process (Kirchenbauer et al., 2023). Zero-shot detectors instead identify generated text from its statistical properties without training on labeled detection examples. DetectGPT (Mitchell et al., 2023) and FAST-DETECTGPT (Bao et al., 2024) use probability-curvature signals, while BINOCULARS compares predictions from two language models (Hans et al., 2024). Supervised detectors learn from labeled human and AI-generated text, as in the fine-tuned Longformer (Beltagy et al., 2020) detector released with MAGE (Li et al., 2024).

Scientific-review detection. Scientific-review detection can exploit both review-specific linguistic patterns and the relationship between a review and the manuscript it evaluates. Liang et al. (2024) estimate AI use at the corpus level from word-frequency changes, while Kumar et al. (2024) use token-frequency patterns to classify individual reviews. ANCHOR generates an AI review of the same paper as a reference and compares it with the target review using whole-review embedding similarity (Yu et al., 2026). A complementary approach places hidden instructions in manuscripts to induce detectable patterns in AI-generated reviews (Rao et al., 2025).

AI-assisted editing further complicates authorship attribution, because human judgements may be expressed in AI-revised language. EDITLENS estimates the extent of AI editing in a text (Thai et al., 2026). SEM-DETECT combines claim-level semantic statistics with textual features to distinguish human-written, AI-rewritten, and fully AI-generated reviews (Duarte et al., 2026). Building on this claim-level evidence, PEERREVEAL retains individual claim representations and learns their interactions instead of reducing their comparisons to predefined aggregate statistics.

Generalization and adversarial training. MAGE evaluates detection across text domains and generator models (Li et al., 2024), while RAID broadens evaluation to diverse models, domains, decoding settings, and adversarial modifications (Dugan et al., 2024). SEM-DETECT also evaluates transfer across writer models, prompting strategies, and review domains. We extend this evaluation setting by automatically searching for diverse ways to generate AI reviews and rewrite human reviews, constructing a broader test set across models and scientific domains.

DIPPER shows that paraphrasing can evade AI-text detectors while largely preserving meaning (Krishna et al., 2023). For robustness to paraphrasing, RADAR trains a paraphraser to evade detection and a detector to recognize its outputs (Hu et al., 2023). Our framework uses detector feedback to optimize a small local review generator and improves three-class detection through training on difficult AI-generated and AI-rewritten reviews under review-quality constraints.

3 PEERSHIFT: A CHALLENGING PEER-REVIEW TEST SET

A detector may perform well under fixed generation procedures yet remain unreliable under other plausible ways of generating AI reviews or rewriting human ones. In practice, the model and workflow used to produce a review are unknown.

We therefore construct PeerShift, a three-class peer-review authorship test set, through automatic search over review-production mechanisms that guide assessment formation or human-review reconstruction. The search selects complementary mechanisms producing challenging, high-quality reviews and fixes them, within each scientific domain, for generation across writer models.

3.1 REVIEW-PRODUCTION MECHANISMS

A review-production mechanism specifies how a writer model forms a review: the writer model is the language model that produces the text, whereas the mechanism determines which evidence it considers and how it reasons toward a judgement. The search below varies the mechanism while holding the base review prompt fixed. We represent a mechanism as a program: an ordered sequence of compatible operations, each a reusable reviewing instruction from a shared library.

For fully AI-generated reviews, operations cover evidence selection, critique and judgement formation, and grounding repair. For example, a program combines operations to select a result, assess alternative explanations against evidence, and form a judgement reflecting uncertainty.

For AI-rewritten reviews, operations vary organization, emphasis, judgement placement, and argumentative presentation while preserving the source's reasons, judgement, and stance. Programs are compiled by appending their operation instructions to a fixed review prompt. For AI generation, the writer receives the manuscript; for rewriting, it receives source-order notes extracted from the human review rather than the review itself, and reconstructs a complete review from them.

3.2 QUALITY-CONSTRAINED MECHANISM SEARCH

We search for complementary mechanisms that increase detection difficulty and diversity while satisfying quality and validity constraints:

$$\mathcal{S}^* = \arg \max_{\mathcal{S} \subseteq \mathcal{Z},\, |\mathcal{S}|=K} \sum_{z \in \mathcal{S}} \min_{e \in \mathcal{E}} \big[r_e(\varnothing) - r_e(z) \big] \quad \text{s.t. } Q_z \geq \tau_Q,\ V_z \geq \tau_V \ \ \forall z \in \mathcal{S}. \tag{1}$$

Here $\mathcal{Z}$ is the program space over the operation library, $r_e(z)$ the recall of development detector $e \in \mathcal{E}$ on the target class (AI-generated or rewritten) of pilot reviews from program z, and $\varnothing$ the base prompt on the same papers, writer models, and ratings; the inner minimum D_z is the difficulty of z. U_z measures marginal diversity in programs and outputs relative to archive $\mathcal{A}$ and breaks remaining ties (Appendix A.2). Q_z measures review utility, while V_z measures paper grounding for AI generation or source fidelity for rewriting; τ_Q and τ_V set minimum quality and validity.

Algorithm 1 summarizes the AI-generation search. The frontier $\mathcal{F}$ contains candidates for expansion, and $\mathcal{A}$ retains those passing the quality and validity checks. ADD introduces an operation, REPLACE substitutes one, COMBINE merges compatible programs, and REPAIR corrects diagnosed validity failures. Candidates are evaluated through pilot reviews generated on a balanced development schedule $\mathcal{D}$ of papers, writer models, and ratings.

Algorithm 1 Automatic quality-constrained mechanism search

Require: prompt p_0; operations $\mathcal{O}$; development schedule $\mathcal{D}$; detector panel $\mathcal{E}$; budget B; output size K

$\mathcal{F} \leftarrow \{\varnothing\}$; $\mathcal{A} \leftarrow \varnothing$ ▷ base prompt, no additional operations
for $b = 1, \ldots, B$ **do**
  $\mathcal{P} \leftarrow \textsc{Expand}(\mathcal{F}, \mathcal{A}, \mathcal{O};\ \textsc{Add}, \textsc{Replace}, \textsc{Combine}, \textsc{Repair})$
  for all complete programs $z \in \mathcal{P}$ **do**
    $p_z \leftarrow \textsc{Compile}(p_0, z)$; $R_z \leftarrow \textsc{BalancedGenerate}(p_z, \mathcal{D})$
    $(D_z, U_z, Q_z, V_z) \leftarrow \textsc{Evaluate}(R_z, \mathcal{E}, \mathcal{A})$; retain z in $\mathcal{A}$ if $Q_z \geq \tau_Q$ and $V_z \geq \tau_V$
  end for
  $\mathcal{F} \leftarrow \textsc{SelectFrontier}(\mathcal{A}; \{D_z, U_z\}_{z \in \mathcal{A}})$
end for
$\mathcal{A} \leftarrow \textsc{Confirm}(\mathcal{A}, \mathcal{D})$ ▷ larger balanced pilots
$\mathcal{S}^* \leftarrow \textsc{SelectComplementary}(\mathcal{A}, K)$; **return** $\textsc{Freeze}(\mathcal{S}^*)$

3.3 CROSS-DOMAIN TEST-SET CONSTRUCTION

The search runs separately for each domain and selects four AI-generation and six rewrite mechanisms for ML and for Nature Communications, and three and five for PeerJ. We freeze them and generate reviews from independent manuscript–human-review corpora, requiring reasoned judgements, rating agreement, and the validity checks above. PeerShift contains 5,430 reviews: 1,830 in its ML subset (PeerShift-ML) and 1,800 each in Nature Communications and PeerJ Psychology and Social Sciences, with balanced classes in each domain. Appendix A details the search controller and detectors, operations, frozen mechanisms and prompts, and dataset composition.

4 THREE-CLASS DETECTOR DESIGN AND ADVERSARIAL TRAINING

After defining the evaluation shifts, we turn to the trainable components of our approach. We first introduce PEERREVEAL's three-class detector architecture and then describe the adversarial training framework. Regular training produces PEERREVEAL-BASE; continuing training with regular and adversarial reviews produces PEERREVEAL.

4.1 PEERREVEAL THREE-CLASS DETECTOR

PEERREVEAL jointly models claim-level semantic evidence and review-level textual signals. It retains the variable-length target and reference claim sequences rather than reducing their relationships to fixed aggregate statistics. Directed self-attention learns from interactions among individual claims and their comparisons to distinguish human-written, AI-rewritten, and fully AI-generated reviews.

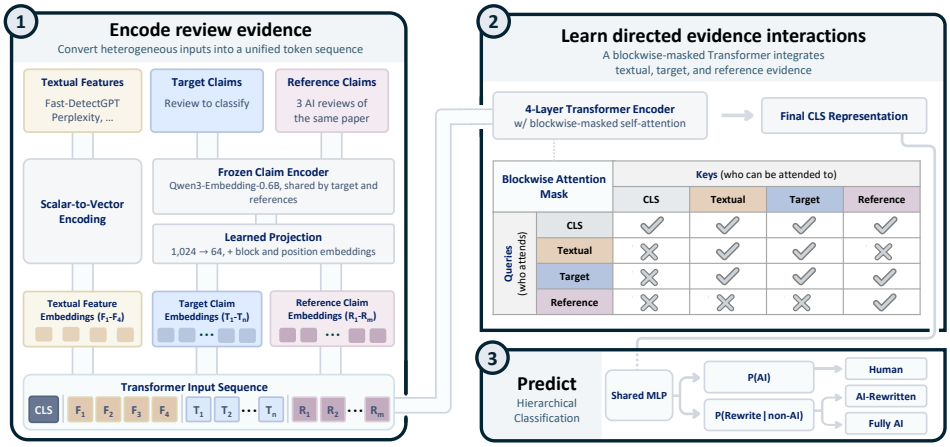


Figure 2: PEERREVEAL detector. A frozen claim encoder and shared projection encode target and reference claims; four textual statistics are encoded as feature vectors. Directed blockwise attention integrates these inputs, and the final CLS representation predicts the three review sources.

Input construction. For each target review, we use three pre-generated AI reviews of the same manuscript and rating as references, following SEM-DETECT (Duarte et al., 2026). We decompose each review into functionally labeled claims and retain evaluation, constructive-input, and clarification-dialogue claims, corresponding to judgements, actionable suggestions, and questions. Each retained claim is represented using a frozen claim encoder. We also retain SEM-DETECT's four review-level textual features for the target: perplexity, entropy, top-200 token percentage, and the FAST-DETECTGPT (Bao et al., 2024) score.

Claim and textual-feature representation. Let $T = (t_1, \ldots, t_n)$ be the retained target-claim sequence, and let $R = (r_1, \ldots, r_m)$ contain the retained claims from the three references. Claims are concatenated in the fixed retained-category order while preserving extractor order within each category. References are concatenated into one pooled block in the frozen reference-selection order.

Claim embeddings share a learned projection into the model space and receive target/reference block and positional embeddings. Let $\hat{z}_l$ be the standardized textual feature $l \in \{1, \ldots, 4\}$. Its token F_l combines a learned feature-identity vector e_l^{id}, the value along a learned value vector e_l^{value}, and the shared textual-block embedding b_{text}: $F_l = e_l^{\text{id}} + \hat{z}_l e_l^{\text{value}} + b_{\text{text}}$. Denoting the resulting target and reference vectors by T_i and R_j, respectively, the encoder input is

$$X = [\text{CLS}, F_1, \ldots, F_4, T_1, \ldots, T_n, R_1, \ldots, R_m]. \tag{2}$$

Claim sequences are padded within each mini-batch, with padding excluded from attention.

Directed blockwise-masked Transformer. An encoder-only Transformer (Vaswani et al., 2017) applies multi-head self-attention under one fixed directed block mask (Figure 2).

The reference block attends only within the pooled references, so its representation is not conditioned on the target review. The target block is the central integration space: it reads target claims, textual features, and reference evidence. Textual-feature tokens attend within their block and to target claims, so that review-level features are interpreted with review content. The CLS token reads itself and all three input blocks to form the final review representation; no other block reads CLS.

Classification and training. The final normalized CLS representation is passed through a shared MLP and two classification heads. The first predicts $a = P(\text{AI})$; the second predicts $b = P(\text{Rewrite} \mid \text{non-AI})$. The resulting three-class distribution is

$$P(\text{AI}, \text{Rewrite}, \text{Human}) = [a, (1-a)b, (1-a)(1-b)]. \tag{3}$$

Training combines binary cross-entropy for AI/non-AI with conditional rewrite/human cross-entropy on non-AI examples. Claim extraction and claim embedding remain fixed; the feature-token parameters, claim projection, block and position embeddings, Transformer, and heads are optimized jointly. Appendices C and E give attention-mask ablations and implementation details, respectively.

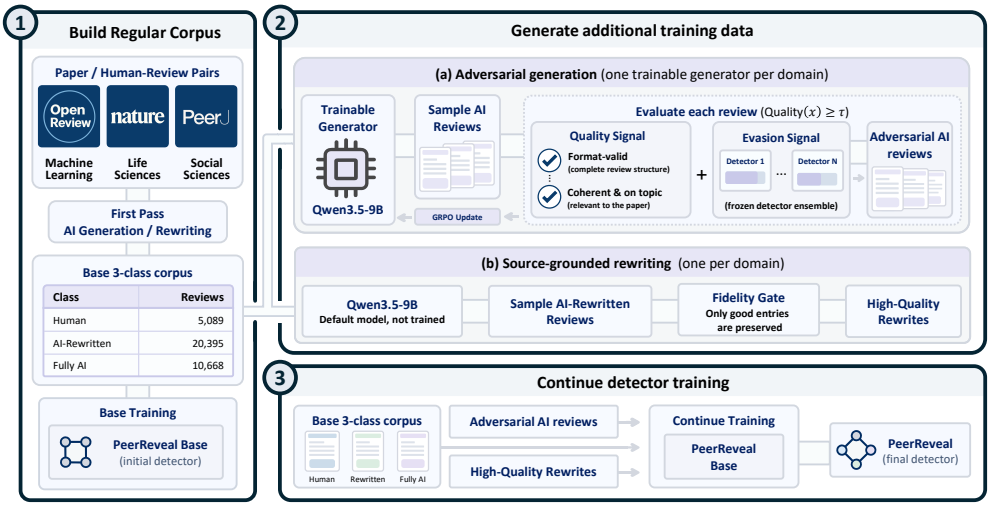


Figure 3: Adversarial generator-detector training. Frozen detector and quality feedback optimize Qwen3.5-9B through GRPO. Three-domain regular training yields PEERREVEAL-BASE; adding adversarial AI reviews and source-grounded rewrites yields PEERREVEAL.

4.2 ADVERSARIAL TRAINING FRAMEWORK

To train PEERREVEAL-BASE, we construct a regular three-class training corpus across ML, Nature Communications, and PeerJ. Each domain supplies manuscript–human-review pairs; conventional prompts generate AI reviews and rewrites. Appendix D.1 gives corpus sizes and generation settings.

We optimize local Qwen3.5-9B (Qwen Team, 2026) generators, one per domain, using review-quality feedback and feedback from a frozen detector ensemble $\mathcal{E}_{\text{fixed}}$, comprising SEM-DETECT, PEERREVEAL, RADAR, and ANCHOR. Their adversarial AI reviews, together with source-grounded rewrites, augment the regular corpus to train PEERREVEAL. Figure 3 summarizes this framework.

4.3 ADVERSARIAL AI REVIEW GENERATOR

4.3.1 STRATEGY-CONDITIONED EXPLORATION

Given the same paper, requested rating, and strategy condition, Qwen3.5-9B independently generates four reviews, which form a comparison group for the GRPO (Shao et al., 2024) update. The three conditions are **paper-only**, which reviews the complete manuscript directly; **human-frame**, which supplies sentence templates taken from real human reviews with their academic vocabulary blanked out and asks the writer to fill in the blanks from the manuscript; and **cached external reviews**, which supplies two same-paper reviews collected in advance and asks the writer to fuse them into one more human-sounding review. Strategy conditioning diversifies the evidence attended to by the generator, the organization of the review, and the formation of critiques and judgements.

4.3.2 DETECTOR FEEDBACK AND GRPO OPTIMIZATION

Each sampled review is evaluated for format validity, review quality, and detector evasion. These evaluations provide the reward signals for GRPO optimization. We combine evasion scores from multiple detectors to guide the generator toward reviews that are difficult across detection methods. Let $e_d(r_i) \in [0, 1]$ denote detector d's evasion score for review r_i, with larger values indicating greater evasion. The joint score is

$$E(r_i) = \left(\sum_{d \in \mathcal{E}_{\text{fixed}}} w_d \sqrt{e_d(r_i)} \right)^2, \qquad \sum_d w_d = 1. \tag{4}$$

The feedback combines the three-class detectors' human/rewrite evidence, RADAR's human probability, and ANCHOR's reference-based decision. Weights are .40, .30, .15, and .15 for PEERREVEAL, SEM-DETECT, RADAR, and ANCHOR. Scorers stay frozen throughout optimization.

Within each group, reviews satisfying the format, quality, and length requirements receive group-relative evasion advantages. Advantages subtract the group mean evasion score of eligible reviews, and GRPO increases the relative likelihood of higher-scoring reviews. Appendices B.1 and B.3–B.4 detail calibration mappings, failure handling, and optimization.

4.4 Source-Grounded Adversarial Rewrite Generation

In the rewrite path, Qwen3.5-9B rewrites each human review given its paper abstract. It preserves the source judgement and reasons while exploring changes in expression, organization, and emphasis. Frozen detector scores guide candidate selection toward rewrites that evade correct classification.

Candidates enter detector training after passing a fidelity gate for judgement, reasons, and stance and a source-distance gate that removes near-copies. The paper abstract supplies grounding for references implicit in the source review. The AI-generation and rewrite paths provide complementary examples for three-class detector training.

4.5 Adversarial Detector Update

From PeerReveal-Base, we continue training on regular reviews and adversarial AI and rewrite reviews from all domains to obtain PeerReveal, sampling by domain and class. Retaining regular data preserves supervision for all three classes while the new examples broaden the distribution.

5 Experiments

5.1 Implementation and Evaluation Setup

Data and protocols. We run two comparisons. The ML-only comparison follows the regular training setting of Sem-Detect (Duarte et al., 2026) and evaluates on four ML test sets: Original, which matches the training writer models and prompts; OOD-M, which uses unseen writer models; OOD-M+P, which uses unseen writer models and prompts; and PeerShift-ML. The main comparison trains on all three domains and evaluates on the full 5,430-review PeerShift test set from Section 3. Nature Communications supplies open peer-review records; PeerJ supplies matched initial manuscripts and first-round human reviews in Psychology and Social Sciences. The three-domain regular corpus contains 36,152 reviews and adversarial augmentation adds 8,464; training, development, and test papers are disjoint, and model selection uses development sets generated with the same mechanisms as PeerShift. Appendices D.1 and D.3 detail the corpora, development sets, and training settings.

Metrics. For binary evaluation, fully AI-generated reviews are positive and the other two classes negative; we report AUROC and true-positive rate (TPR) at 1% false-positive rate (FPR), both from continuous AI scores ($P(\mathrm{AI})$ for three-class detectors), and binary Macro-F1 at a decision threshold fixed before evaluation (Appendix D.6). Three-class evaluation reports Macro-F1 and class recall. Pooled metrics combine all 5,430 predictions rather than averaging domain scores.

5.2 Baselines

We compare general-purpose detectors (LogRank (Ippolito et al., 2020), Fast-DetectGPT (Bao et al., 2024), Binoculars (Hans et al., 2024), Mage (Li et al., 2024), Radar (Hu et al., 2023), and EditLens (Thai et al., 2026)) and review-specific methods (Tf (Kumar et al., 2024), Anchor (Yu et al., 2026), and Sem-Detect (Duarte et al., 2026)). Anchor uses same-paper GPT-5.5 reference reviews. Sem-Detect is trained on the same regular corpus as PeerReveal-Base and, for a matched comparison with PeerReveal, also on the augmented corpus.

We also fine-tune RoBERTa (Liu et al., 2019) on full reviews with a three-class classification head, using the same regular and augmented training corpora.

5.3 Research Questions

Our experiments ask: (1) how challenging PeerShift is for existing detectors (Section 6.1); (2) how well PeerReveal distinguishes three review sources across domains and improves with adversar-

ial training (Section 6.2); (3) how PEERREVEAL compares with SEM-DETECT under its original training and test settings (Section 6.3); and (4) whether adversarial training enables small local 9B generators to evade multiple detectors (Section 6.4).

6 RESULTS

6.1 EXISTING DETECTORS STRUGGLE ON PEERSHIFT

Table 1 evaluates whether detectors recognize fully AI-generated judgement without flagging human judgement expressed with AI assistance, with human-written and AI-rewritten reviews as negatives. PEERREVEAL achieves the highest AUROC (98.02%) and binary Macro-F1 (93.02%), while adversarially trained ROBERTA leads in TPR at 1% FPR (72.76%; Figure 4). For PEERREVEAL, PeerJ is the hardest domain at low FPR (TPR at 1% FPR: 49.83%, versus 77.54% on ML and 50.50% on Nature Communications); per-domain results and ROC curves are in Appendix D.6.

Table 1: Pooled binary detection on PeerShift (%). Positive: AI-generated; negative: human-written and AI-rewritten. SEM-DETECT and ROBERTA use matched regular/adversarial training corpora.

Detector	AUROC	TPR@1%	Macro-F1
LOGRANK	35.39	0.88	40.03
FAST-DETECTGPT	64.59	19.23	63.99
BINOCULARS	44.24	0.00	49.39
RADAR	67.77	1.10	59.50
ANCHOR	87.26	25.91	74.14
TF	55.52	6.19	50.98
MAGE	52.73	0.11	39.99
EDITLENS	73.11	0.11	63.21
ROBERTA (regular)	77.59	49.72	67.66
ROBERTA (adversarial)	89.10	**72.76**	83.76
SEM-DETECT (regular)	77.87	27.35	40.00
SEM-DETECT (adversarial)	87.13	18.73	74.59
PEERREVEAL-BASE	92.32	34.86	82.80
PEERREVEAL	**98.02**	60.77	**93.02**

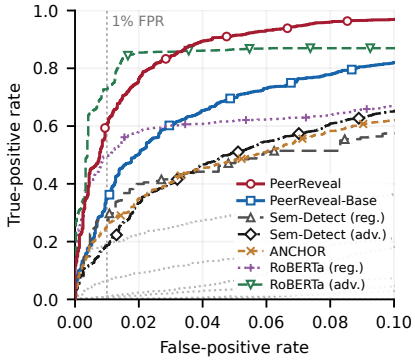


Figure 4: Pooled ROC curves on PeerShift (FPR 0–0.1). Weaker baselines are gray. Appendix D.6 gives full-range and per-domain ROC curves.

6.2 PEERREVEAL IMPROVES THREE-CLASS DETECTION, WITH FURTHER GAINS FROM ADVERSARIAL TRAINING

Table 2 compares SEM-DETECT, ROBERTA, and PEERREVEAL under matched regular and adversarial training. With regular training, PEERREVEAL-BASE reaches 70.52% pooled Macro-F1, 11.48 points above ROBERTA; adversarial training with reviews from local 9B generators adds 11.88 points, bringing PEERREVEAL to 82.40%, with the largest gain on ML (+21.22). ROBERTA also benefits from it, reaching 73.04%, while PEERREVEAL remains 9.36 points ahead.

Table 2: Three-class Macro-F1 (%) with matched corpora; reg./adv. denote regular/augmented training.

	PeerShift			
Detector	ML	Nature	PeerJ	Pooled
ROBERTA (reg.)	57.01	58.54	61.07	59.04
ROBERTA (adv.)	69.92	71.31	77.56	73.04
SEM-DETECT (reg.)	39.84	55.09	58.30	51.52
SEM-DETECT (adv.)	51.52	55.59	60.20	56.17
PEERREVEAL-BASE	58.73	73.93	77.67	70.52
PEERREVEAL	**79.95**	**81.98**	**85.18**	**82.40**

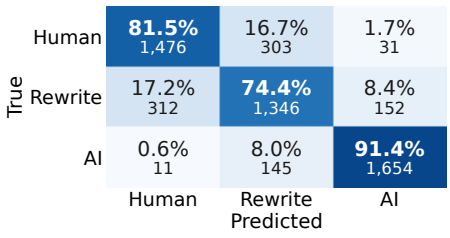


Figure 5: PEERREVEAL three-class confusion matrix on pooled PeerShift (% by row). See Appendix D.5 for class recalls.

Class-wise performance. PEERREVEAL recalls 91.38% of AI-generated, 74.36% of rewritten, and 81.55% of human reviews (Figure 5). Human reviews are almost never labeled AI-generated (1.71%); the errors lie on the human/rewrite boundary, where 16.74% are predicted as rewrites. Appendix F works through one paper's three reviews with every detector's output.

6.3 PEERREVEAL OUTPERFORMS SEM-DETECT UNDER ITS ORIGINAL TRAINING AND TEST SETTINGS

Under SEM-DETECT's ML-only regular training (Table 3), PEERREVEAL-BASE leads by 11.83 points on Original, 21.55 on OOD-M, and 18.84 on OOD-M+P. Its 48.14% on PeerShift-ML marks the remaining generalization gap under broader review-generation conditions (Appendix D.2).

Table 3: Comparison under SEM-DETECT's ML-only regular training: pooled Macro-F1 (%).

Detector	Original	OOD-M	OOD-M+P	PeerShift-ML
SEM-DETECT	84.14	71.59	69.18	34.40
PEERREVEAL-BASE	**95.97**	**93.14**	**88.01**	**48.14**
Gain (pp)	+11.83	+21.55	+18.84	+13.74

6.4 ADVERSARIALLY TRAINED LOCAL 9B GENERATORS EVADE MULTIPLE DETECTORS

Table 4 evaluates the local Qwen3.5-9B (Qwen Team, 2026) generators before and after adversarial training. Evasion is the fraction of quality-eligible reviews classified as non-AI on held-out papers (Appendix B.5). Training increases evasion for most detectors, including EDITLENS and TF, neither used for feedback. PEERREVEAL is less evaded than PEERREVEAL-BASE at both thresholds; Appendix G shows an adversarial review it detects. These local 9B generators enable effective and practical adversarial training (Table 2), with gains that transfer to PeerShift's mechanisms and writers absent from generator training.

Table 4: Pooled evasion (%) before/after generator training at 1% and 5% human FPR, among quality-eligible reviews. † marks detector families used for generator-training feedback. We omit weaker detectors exceeding 91% evasion before training at 1% FPR; Appendix B.5 gives all details.

Detector	1% FPR		5% FPR	
	Before	After	Before	After
RADAR†	98.80	100.00	85.29	78.55
ANCHOR†	76.28	99.67	41.74	95.05
TF	88.29	94.06	58.26	76.90
EDITLENS	59.76	88.45	0.30	40.59
ROBERTA (regular)	12.91	43.89	10.21	30.03
ROBERTA (adversarial)	0.90	42.90	0.90	42.57
SEM-DETECT (regular)†	79.88	100.00	61.56	100.00
PEERREVEAL-BASE†	55.26	99.01	18.02	95.71
PEERREVEAL†	6.31	72.28	0.00	48.84

7 CONCLUSION

PEERREVEAL combines learned claim interactions with adversarial supervision from a local Qwen3.5-9B generator. Quality-constrained mechanism search produces PeerShift, covering diverse review-production processes across three domains. Under matched corpora, PEERREVEAL outperforms SEM-DETECT and ROBERTA in three-class detection under both training regimes, reaching 82.40% pooled Macro-F1 on 5,430 reviews. These results show the value of learning from individual claims and diverse training reviews.

AI USE STATEMENT

Generative AI tools were used to assist with experiment orchestration, software development, data processing, and drafting and editing parts of the manuscript. The authors reviewed all AI-assisted work and take responsibility for the final scientific content, claims, analyses, and artifacts.

ETHICS STATEMENT

PEERREVEAL is intended to help editors and program chairs understand review provenance, not to adjudicate individual cases. Its outputs are probabilistic, its errors concentrate on the boundary between human-written and AI-rewritten reviews, and false accusations carry real costs for reviewers; detector predictions should therefore serve as one signal among several rather than as grounds for action on their own. All human reviews come from venues with public review records, reviewer identities are not used, and generated and rewritten reviews are used only for training and evaluation. Adversarial generators pose dual-use risks; here they serve only to improve detection.

REPRODUCIBILITY STATEMENT

Appendix A gives the mechanism-search controller, the operation families, the numerical acceptance constraints, the frozen mechanisms selected for each domain, and their exact prompt texts; Appendix B gives the detector-score calibration, the quality judge with its rubric and prompt, GRPO settings, and the generator evaluation protocol; Appendix C gives all 15 attention structures with their results; Appendix D gives corpus composition, paper-level splits, training and model-selection settings, baseline implementations, the fixed decision rule for binary metrics, per-class results, human-versus-AI binary results, and generation and detection cost; Appendix E gives the frozen claim-extraction, reference-selection, and encoding pipeline together with every detector hyperparameter; and Appendix F gives one paper's three reviews in full with every detector's output on them. Final training configurations and execution commands accompany the code.

REFERENCES

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 24814–24836, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6b8c6f846c3575e1d1ad496abea28826-Paper-Conference.pdf.

Paul Barde, Tristan Karch, Derek Nowrouzezahrai, Clément Moulin-Frier, Christopher Pal, and Pierre-Yves Oudeyer. Learning to Guide and to be Guided in the Architect-Builder Problem. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=swiyAeGzFhQ.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense X Retrieval: What Retrieval Granularity Should We Use? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15159–15177, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.845. URL https://aclanthology.org/2024.emnlp-main.845/.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

DeepSeek-AI. DeepSeek-V3 Technical Report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin et al. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models, 2025. URL https://arxiv.org/abs/2512.02556.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong et al. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

André V. Duarte, João DS Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L. Oliveira. LumberChunker: Long-Form Narrative Document Segmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6473–6486, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.377. URL https://aclanthology.org/2024.findings-emnlp.377/.

Andre Vicente Duarte, Brian Tufts, Aditya Oke, Fei Fang, Arlindo L. Oliveira, and Lei Li. SemDetect: Semantic Level Detection of AI Generated Peer-Reviews. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=3S9eVockjL.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.674. URL https://aclanthology.org/2024.acl-long.674/.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang et al. GLM-5: from Vibe Coding to Agentic Engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. RADAR: Robust AI-Text Detection via Adversarial Learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 15077–15095. Curran Associates, Inc., 2023. doi: 10.52202/075280-0662. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/30e15e5941ae0cdab7ef58cc8d59a4ca-Paper-Conference.pdf.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.164. URL https://aclanthology.org/2020.acl-main.164/.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier et al. Mistral 7B, 2023. URL https://arxiv.org/abs/2310.06825.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.,

2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A Watermark for Large Language Models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 27469–27500. Curran Associates, Inc., 2023. doi: 10.52202/075280-1195. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/575c450013d0e99e4b0ecf82bd1afaa4-Paper-Conference.pdf.

Sandeep Kumar, Mohit Sahu, Vardhan Gacche, Tirthankar Ghosal, and Asif Ekbal. 'Quis custodiet ipsos custodes?' Who will watch the watchmen? On Detecting AI-generated peer-reviews. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22663–22679, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1262. URL https://aclanthology.org/2024.emnlp-main.1262/.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. MAGE: Machine-generated Text Detection in the Wild. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–53, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.3. URL https://aclanthology.org/2024.acl-long.3/.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang et al. Monitoring AI-Modified Content at Scale: A Case Study on the Impact of ChatGPT on AI Conference Peer Reviews. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29575–29620. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liang24b.html.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. URL https://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Quang Pham, Chenghao Liu, and Steven C. H. Hoi. Continual Normalization: Rethinking Batch Normalization for Online Continual Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=vwLLQ-HwqhZ.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Vishisht Srihari Rao, Aounon Kumar, Himabindu Lakkaraju, and Nihar B. Shah. Detecting LLM-generated peer reviews. *PLOS ONE*, 20(9):1–22, 09 2025. doi: 10.1371/journal.pone.0331871. URL https://doi.org/10.1371/journal.pone.0331871.

Abdelrahman Sadallah, Tim Baumgärtner, Iryna Gurevych, and Ted Briscoe. The Good, the Bad and the Constructive: Automatically Measuring Peer Review's Utility for Authors. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28991–29021, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1476. URL https://aclanthology.org/2025.emnlp-main.1476/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen et al. Kimi K2.5: Visual Agentic Intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Katherine Thai, Bradley Emi, Elyas Masrour, and Mohit Iyyer. EditLens: Quantifying the Extent of AI Editing in Text. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 122997–123024, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c7d936e696f57468b6b80a7b1463178c-Paper-Conference.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions. *Computational Linguistics*, 51(1):275–338, March 2025. doi: 10.1162/coli_a_00549. URL https://aclanthology.org/2025.cl-1.8/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv et al. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

Sungduk Yu, Man Luo, Avinash Madasu, Vasudev Lal, and Phillip Howard. Is Your Paper Being Reviewed by an LLM? Benchmarking AI Text Detection in Peer Review. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 61517–61553, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/64592277a1cbb02ed6e26b6e9e5f458c-Paper-Conference.pdf.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin et al. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176*, 2025.

Li Zhou, Ruijie Zhang, Xunlian Dai, Daniel Hershcovich, and Haizhou Li. ScholarLens: Tracking the growing penetration of Large Language Models in scholarly writing and peer review. *Engineering Applications of Artificial Intelligence*, 177:114988, August 2026. ISSN 0952-1976. doi: 10.1016/j.engappai.2026.114988. URL http://dx.doi.org/10.1016/j.engappai.2026.114988.

A REVIEW-MECHANISM SEARCH DETAILS

A.1 FULL SEARCH CONTROLLER

Algorithm 2 expands the compact controller in Algorithm 1. It records every complete-program evaluation and uses the trace when proposing replacements, combinations, and failure-driven repairs.

Algorithm 2 Full automatic quality-constrained mechanism-search controller

Require: base prompt p_0; class-specific operation registry $\mathcal{O}$; balanced development schedule $\mathcal{D}$; detector panel $\mathcal{E}$; search budget B; output size K

$\mathcal{F} \leftarrow \{\varnothing\}$; $\mathcal{A} \leftarrow \varnothing$; $\mathcal{T} \leftarrow \varnothing$
for $b = 1, \ldots, B$ **do**
  $\mathcal{P} \leftarrow \textsc{Propose}(\mathcal{F}, \mathcal{A}, \mathcal{T}, \mathcal{O};\ \textsc{Add}, \textsc{Replace}, \textsc{Combine}, \textsc{Repair})$
  for all complete programs $z \in \mathcal{P}$ **do**
    $p_z \leftarrow \textsc{Compile}(p_0, z)$; $R_z \leftarrow \textsc{BalancedGenerate}(p_z, \mathcal{D})$
    $(D_z, Q_z, V_z) \leftarrow \textsc{Evaluate}(R_z, \mathcal{E})$
    $U_z \leftarrow \textsc{MarginalDiversity}(z, R_z, \mathcal{A})$
    append (z, D_z, U_z, Q_z, V_z) to trace $\mathcal{T}$
    if $Q_z \geq \tau_Q$ and $V_z \geq \tau_V$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{z\}$
    end if
  end for
  $\mathcal{F} \leftarrow \textsc{SelectFrontier}(\mathcal{P}, \mathcal{A}; \{D_z, U_z\}_{z \in \mathcal{P}})$
end for
$\mathcal{A} \leftarrow \textsc{Confirm}(\mathcal{A}, \mathcal{D})$ ▷ larger balanced pilots
$\mathcal{S}^* \leftarrow \textsc{SelectComplementary}(\mathcal{A}, K; \{D_z, U_z\}_{z \in \mathcal{A}})$
return $\textsc{Freeze}(\mathcal{S}^*)$ and trace $\mathcal{T}$

$\mathcal{F}$ is the search frontier, $\mathcal{A}$ contains quality- and validity-qualified programs, and $\mathcal{T}$ stores their complete evaluations. ADD introduces an operation, REPLACE changes one operation, COMBINE merges compatible programs, and REPAIR corrects diagnosed validity failures. Difficulty and marginal diversity determine frontier promotion and final complementary selection.

Development detectors. The ML search uses four frozen three-class detectors: a fine-tuned RoBERTa-large (Liu et al., 2019), SEM-DETECT (Duarte et al., 2026), and two auxiliary claim-based models, Target-only and Pairwise. Nature Communications and PeerJ use the latter three. This panel covers whole-review classification, predefined semantic and textual features, and learned claim representations. RoBERTa averages chunk representations before a three-class linear head; SEM-DETECT uses its nine-feature LightGBM classifier. The auxiliary models operate on frozen 1,024-dimensional claim embeddings and are described below. Both auxiliary models are trained on the ML regular corpus using an 85/15 training–validation split and remain frozen during search.

Target-only. Each target claim is projected to 64 dimensions with layer normalization and GELU, then encoded by a 128-unit feed-forward layer with ReLU and dropout. A learned query scores the claim vectors; softmax-normalized scores weight their sum into a review representation. An MLP with a 128-unit hidden layer outputs three-class logits. This model uses neither reference claims nor textual features.

Pairwise. Separate projections map target and reference claims to 64 dimensions. A pair-scoring MLP with 64 hidden units takes the two vectors, their elementwise absolute difference, and their elementwise product. For each target claim, softmax over these scores forms a weighted reference vector. The target vector, weighted reference vector, their absolute difference and product, and the maximum pair score are encoded into a 128-dimensional claim representation. Learned-query attention pooling then summarizes the target claims. A separate 32-unit feed-forward layer encodes the four textual features; its output is concatenated with the pooled claim representation and passed to a 128-unit MLP with a three-class output.

A.2 Acceptance Constraints

The thresholds τ_Q and τ_V of Section 3 are fixed before the search. A candidate program joins $\mathcal{A}$ only if all of the following hold over its pilot reviews: at least 85% pass the quality judge of Appendix B.2; at least 95% of the pilot reviews are distinct; and the program reduces target-class recall relative to the matched control for at least two development detectors. For ML the gate additionally requires that the median review length and the median retained-claim count each reach 60% of the matched human median. For Nature Communications and PeerJ it instead requires that at least 85% of the pilot reviews pass the class-specific validity check.

Validity is checked per class. A fully AI-generated review passes when it is paper-grounded, contains a judgement and a supporting reason, is consistent with the requested rating, and is on topic. A rewrite passes when it preserves the source review's central judgement, main reasons, and stance, contains a judgement and a supporting reason, is an independent reconstruction, adds no substantive claim, and is on topic. Empty outputs and outputs carrying prompt residue are rejected before either check.

In the ML search, promotion from pilot to confirmation additionally requires, on a larger balanced pilot, that at least three of the four development detectors lose recall, that no detector gain more than $.05$ recall, and that the median recall reduction reach $.10$.

Among qualified programs, difficulty is ranked lexicographically by the minimum per-detector recall reduction relative to the matched control, then the median reduction, then the negated worst-case absolute recall; among equally difficult programs, the one with less operation overlap with those already selected is preferred. Per-detector recalls, mean AI probabilities, and error directions are recorded for every evaluation.

A.3 Class-Specific Operation Spaces

AI operations span four families: **evidence selection** over assumptions, baselines, results, conditions, scope, and limitations; **critique formation** via dependency analysis, baseline–question alignment, alternative interpretation, consequence reasoning, and counterfactuals; **judgement formation** linking evidence to confidence, interpretation, or recommendation; and **grounding repair** preserving metric direction, experimental conditions, evidence scope, and strength–boundary consistency.

The rewrite registry begins from source-order bullet notes extracted from the human review (Appendix A.5), which record its central judgement, reasons, and stance. Operations vary organization, emphasis, judgement placement, prose and section structure, argumentative expansion, and connections among the original reasons. Failure-driven repair removes direct paraphrasing, incomplete reasoning, unsupported additions, and loss of source stance.

A.4 Selected Mechanisms

The search runs separately for each domain over the shared operation library, so the selected mechanisms differ across domains. Table 5 lists the selected AI-generation mechanisms and Table 6 the selected rewrite mechanisms, together with the number of PeerShift reviews each mechanism produced. Appendix A.5 gives the exact prompt text of every mechanism.

A.5 Frozen Prompt Texts

AI generation. Every AI-generated review uses one base prompt, which fixes the venue and the requested rating; the requested rating is taken from the human reviews of the same paper, and the paper text is supplied with the prompt:

> Review the given paper for *[a top AI conference | a Nature Communications journal submission | a PeerJ Psychology / Social Sciences journal submission]*. Be critical, focused, constructive, and paper-grounded, and make the final recommendation exactly "*[rating]*." Give a complete assessment of the contribution, supported strengths, substantive weaknesses or limitations, clarification questions where they matter, and the reasoning behind the judgement. Organize the review as natural reviewer prose without a prescribed five-part heading scaffold.

Table 5: Selected AI-generation mechanisms by domain.

Domain	ID	Reasoning-operation composition	Reviews
ML	ML-A1	Assumption dependency + alternative result interpretations	152
	ML-A2	Baseline–question fit + alternative result interpretations + observation–consequence reasoning	152
	ML-A3	Baseline–question fit + decision counterfactual	153
	ML-A4	Verified same-evidence strength/boundary judgement	153
Nature	NC-A1	Methodological benefit–cost tradeoff	150
	NC-A2	Constructive resolution effect + methodological tradeoff	150
	NC-A3	Claim–evidence clarification questions + alternative result interpretations, with grounding repair	150
	NC-A4	Cross-section claim–evidence consistency + methodological tradeoff	150
PeerJ	PJ-A1	Verified strength–limitation chain, with grounding repair	200
	PJ-A2	Cross-section claim–evidence consistency	200
	PJ-A3	Verified decision counterfactual, with grounding repair	200

A mechanism appends a single block whose operation sentences are the only part that varies:

> Evaluation-reasoning guidance:
> Shape the assessment through these paper-grounded reasoning relations when applicable: *[operation sentences]*. Tie each resulting judgement to evidence that is verifiably present, and treat uncertainty as a scoped question rather than an asserted deficiency. Keep the assessment complete and preserve the exact requested recommendation.

The operation sentences of the selected mechanisms are:

ML-A1 Identify assumptions on which an important conclusion depends; Explain what changes if an assumption is weakened, violated, or left unverified; Reflect that dependency in the assessment; When a reported result admits more than one explanation, distinguish the plausible interpretations; Identify what evidence favors one interpretation over the others; Calibrate the assessment to the remaining ambiguity.

ML-A2 Evaluate whether each important comparison actually tests the question the paper asks; Explain what the baseline can establish and what it cannot; Reflect any mismatch in the interpretation of the reported advantage; When a reported result admits more than one explanation, distinguish the plausible interpretations; Identify what evidence favors one interpretation over the others; Calibrate the assessment to the remaining ambiguity; Begin evaluative points from a concrete observation in the paper; Explain the consequence of that observation for validity, usefulness, or interpretation; Make the consequence proportional to what the observation actually supports.

ML-A3 Evaluate whether each important comparison actually tests the question the paper asks; Explain what the baseline can establish and what it cannot; Reflect any mismatch in the interpretation of the reported advantage; State which verified change in evidence or interpretation would alter the recommendation; Explain why the current record falls on one side of that decision boundary; Keep the counterfactual plausible and specific to the paper.

ML-A4 Before writing the final review, silently verify every material statement against the provided paper; For each decisive comparison, theoretical statement, or experimental interpretation, name its Section, Table, Figure, Equation, or Theorem when one is available; preserve the metric meaning, direction, values, assumptions, and scope exactly; Do not use external facts or assert statistical significance, computational cost, missing baselines, missing experiments, or missing ablations unless the paper explicitly establishes them; If an absence or interpretation cannot be verified, retain the issue as a clarification question rather than a factual assertion; Use the same explicitly reported observation to explain both a genuine strength and the boundary of what that strength supports, and carry that evidence-weighted distinction into the requested recommendation while still providing the complete review requested by the base prompt.

NC-A1 Select a methodological choice that creates a benefit and a cost; Explain whether the benefit-cost tradeoff is appropriate for the paper's stated objective; Tie the methodological tradeoff to the overall assessment.

NC-A2 Select an identified issue for which a concrete author response would matter; Select a methodological choice that creates a benefit and a cost; Explain what a concrete requested change would resolve and

Table 6: Selected rewrite mechanisms by domain, with the reconstruction instruction appended to the shared rewrite prompt (Appendix A.5).

Domain	ID	Appended reconstruction instruction	Reviews
ML	ML-R1	Make the final judgement feel calibrated to the evidence, but keep the wording less polished and recognizably reviewer-like.	102
	ML-R2	Embed the recommendation inside the discussion instead of isolating it. Keep the prose slightly rough but coherent.	102
	ML-R3	Write in less polished but still professional reviewer prose. Keep the judgement and reasons explicit, and avoid generic assistant-like smoothing.	102
	ML-R4	Embed the recommendation in the discussion rather than isolating it. Keep the wording conservative and slightly less polished, while preserving every main reason from the notes.	102
	ML-R5	Reconstruct the review in plain, slightly less polished reviewer prose. Include a clear judgement and reason, but do not add new claims or uplift the recommendation beyond the source notes.	101
	ML-R6	Reconstruct a complete reviewer note from the bullet notes. Use natural but not overly polished prose, preserve the stance and all main reasons, and include one explicit judgement with its reason.	101
Nature	NC-R1	Use less-polished but professional reviewer wording. Use a natural reviewer narrative rather than rubric prose. Use plain or straightforward reviewer prose. Avoid template-like transitions and phrasing.	100
	NC-R2	Use less-polished but professional reviewer wording. Avoid generic assistant-like smoothing.	100
	NC-R3	Use a natural reviewer narrative rather than rubric prose.	100
	NC-R4	Use plain or straightforward reviewer prose. Use less-polished but professional reviewer wording. Avoid generic assistant-like smoothing.	100
	NC-R5	Embed the recommendation inside the discussion.	100
	NC-R6	Embed the recommendation inside the discussion. Use slightly rough but coherent prose.	100
PeerJ	PJ-R1	Place the main judgement before its supporting reasons.	120
	PJ-R2	Use a natural or plain reviewer voice. Use light reviewer hedging. Permit uneven transitions found in reviewer prose. State an explicit judgement together with its reason.	120
	PJ-R3	Preserve mixed confidence and both promising and concerning aspects.	120
	PJ-R4	Preserve mixed confidence and both promising and concerning aspects. Use plain or straightforward reviewer prose.	120
	PJ-R5	Produce a shorter reconstruction while retaining central content. Be concise while retaining substantive judgement and reasoning.	120

why that resolution matters; Explain whether the benefit-cost tradeoff is appropriate for the paper's stated objective; Connect the effect of the requested resolution to the overall assessment; Tie the methodological tradeoff to the overall assessment.

NC-A3 Identify a central position stated by the paper before evaluating it; Select a reported result whose interpretation affects the assessment; Select a specific unresolved interpretation that warrants clarification; Distinguish plausible interpretations of the result and identify what evidence favors one over another; Explain how different answers to the unresolved interpretation would change confidence or judgement; Separate what the evidence resolves from an unresolved issue that should remain a question; Calibrate confidence and recommendation to the ambiguity that remains; Connect the decision-relevant clarification to the requested recommendation; Express an unverifiable absence or interpretation as a clarification question rather than a factual assertion; Do not generalize beyond the evidence or invent an unreported factual, experimental, theoretical, or evaluative claim.

NC-A4 Select a methodological choice that creates a benefit and a cost; Select related claims or evidence stated in different parts of the paper; Explain whether the benefit-cost tradeoff is appropriate for the paper's stated objective; Determine whether the selected statements reinforce one another or create a substantive inconsistency; Tie the methodological tradeoff to the overall assessment; Explain how the consistency or conflict affects confidence and recommendation.

PJ-A1 Select a genuine paper-supported strength whose value depends on a condition or limitation; Explain whether the limitation merely qualifies the strength or substantially undercuts it; Carry the strength-

limitation dependency into the judgement; Preserve metric meaning, direction, values, assumptions, and scope exactly, including reported conditions; Name the relevant Section, Table, Figure, Equation, or Theorem when one is available; Express an unverifiable absence or interpretation as a clarification question rather than a factual assertion; Do not generalize beyond the evidence or invent an unreported factual, experimental, theoretical, or evaluative claim; Use only facts and evidence contained in the provided paper.

PJ-A2 Select related claims or evidence stated in different parts of the paper; Determine whether the selected statements reinforce one another or create a substantive inconsistency; Explain how the consistency or conflict affects confidence and recommendation.

PJ-A3 Identify the paper-supported evidence or interpretation that is decision-critical; State which verified clarification or change in evidence would alter the recommendation; Explain why the current evidence falls on the stated side of the decision boundary; Preserve metric meaning, direction, values, assumptions, and scope exactly, including reported conditions; Name the relevant Section, Table, Figure, Equation, or Theorem when one is available; Express an unverifiable absence or interpretation as a clarification question rather than a factual assertion; Do not generalize beyond the evidence or invent an unreported factual, experimental, theoretical, or evaluative claim; Use only facts and evidence contained in the provided paper.

Rewriting. Rewrite writers do not see the human review. A fixed extraction step (Mistral Large 3) first converts each source review into source-order bullet notes. For Nature Communications and PeerJ the extraction instruction is "Extract source-order notes from one academic peer review. Return only bullet notes. Preserve the central judgement, every main supporting reason, strengths, weaknesses, technical concerns, author questions, stance, and any recommendation. Do not add content." For ML it is "Convert the provided peer review into concise source-order bullet notes. Preserve the reviewer's stance, main points, questions, and judgement if they appear in the source. Do not add new content, do not improve the review, and do not impose section headings. Return only bullet notes." The writer then receives the system prompt "Return only the rewritten peer review text. Do not include reasoning, JSON, markdown fences, instruction summaries, bullet notes, or comments about being an AI model." and the reconstruction prompt:

> The following text is source-order bullet notes extracted from one human peer review. Reconstruct a meaningful rewritten peer review from those notes.
> Hard constraints:
> – Preserve the original reviewer's main judgement, reasons, and stance.
> – Do not add substantive new claims beyond the bullet notes.
> – Do not change the final judgement.
> – Include at least one clear judgement about the paper and the reason for that judgement.
> – Do not use donor human-style profile statistics; do not mention this instruction.

For ML, the mechanism follows as "Style constraint: *[instruction]*" and the bullet notes close the prompt. For Nature Communications and PeerJ, the prompt additionally states "Independently reconstruct the complete review from the notes. Direct paraphrasing or imitation of sentences from the source review is prohibited.", supplies the paper abstract as context, and lists the mechanism's operations as numbered lines under "Reconstruction mechanism:" before the notes. Table 6 gives the appended text of every mechanism.

A.6 PeerShift Composition

Table 7: Composition of PeerShift by domain. Parentheses give the number of writer models and mechanisms behind each generated class.

Domain	Human	Fully AI-generated	AI-rewritten
ML	610	610 (10 writers, 4 mechanisms)	610 (5 writers, 6 mechanisms)
Nature Communications	600	600 (10 writers, 4 mechanisms)	600 (5 writers, 6 mechanisms)
PeerJ	600	600 (10 writers, 3 mechanisms)	600 (5 writers, 5 mechanisms)

The generated classes span distinct writer-model pools, generation procedures, and review styles. Every generated review must contain a reasoned judgement; AI-generated reviews must also match

Table 8: PeerShift generation models and per-domain counts. † links to official documentation.

Class	Writer model	ML	Nature	PeerJ
AI	Gemini 3.1 Pro Preview †	61	60	60
	DeepSeek V4 Pro (DeepSeek-AI et al., 2026)	61	60	60
	MiniMax M2.5 †	61	60	60
	Mistral Large 3 (675B) †	61	60	60
	Qwen3-Next-80B-A3B †	61	60	60
	GLM-5 (GLM-5-Team et al., 2026)	61	60	60
	Kimi K2.5 (Kimi Team et al., 2026)	61	60	60
	Grok 4.3 †	61	60	60
	Claude Sonnet 4.6 †	61	60	60
	GPT-5.5 †	61	60	60
Rewrite	Gemini 3.5 Flash †	122	120	120
	DeepSeek V4 Pro	122	120	120
	Kimi K2.5	122	120	120
	Grok 4.3	122	120	120
	GPT-5.5	122	120	120

the requested rating and pass the paper-grounding check, and rewrites must keep the source judgement and pass the source-fidelity check. Mechanisms are selected and frozen at the program level before balanced full-set generation.

The PeerJ source corpus covers 460 Psychology and 140 Social Sciences papers; its initial manuscripts are paired with first-round human reviews. Across the three subsets, the final set contains 1,810 reviews per class and 5,430 in total.

Table 8 lists generation models and review counts. Writers instantiate the selected reasoning or reconstruction programs; test reviews are separate from those generated locally for detector training.

B AI Generator Training Details

B.1 Detector Score Mappings

The weights in Equation 4 are .30, .40, .15, and .15 for Sem-Detect, PeerReveal, Radar (Hu et al., 2023), and Anchor (Yu et al., 2026), respectively. Sem-Detect's evasion score is $P(\text{Human}) + .3P(\text{Rewrite})$. For PeerReveal, we use the human-class logit margin

$$m_H = \ell_H - \log(\exp(\ell_{\text{AI}}) + \exp(\ell_{\text{Rewrite}}))\,. \tag{5}$$

Three increasing calibration anchors map to 0, .5, and 1; linear interpolation between anchors and clipping outside them produce the evasion score. Radar's human-probability signal is mapped through a fixed empirical cumulative distribution. Anchor provides a binary evasion score at its threshold calibrated to a .01 false-positive rate. The mappings stay fixed throughout optimization.

B.2 Quality Judge

The same frozen judge scores review quality for the mechanism-search acceptance constraints (Appendix A.2), for generator credit assignment, and for generator evaluation. It is Mistral Large 3 (675B)[1] served through Amazon Bedrock at temperature 0 with a budget of 512 output tokens; a JSON schema is supplied in the prompt text and decoding is unconstrained. The judge receives the paper title, the manuscript text truncated to 12,000 characters, and the review truncated to 20,000 characters. It does not see the writer model, the mechanism, the detector scores, or whether the review is a training sample. Its system prompt is:

> You are a strict peer-review quality judge. Your task is to evaluate whether a submitted review is a coherent, useful, paper-specific peer review. You must judge the review as a

[1] https://huggingface.co/mistralai/Mistral-Large-3-675B-Instruct-2512

> review, not as generic academic-sounding text. Penalize text that is repetitive, globally incoherent, semantically thin, corrupted, full of invented or irrelevant terms, or not grounded in the paper. Do not reward length by itself. A concise coherent review is better than a long noisy one. Return only valid JSON.

The judge rates five dimensions from 1 to 5 (Table 9) and returns a validity flag with a one-sentence rationale. Its score averages the first four dimensions; format compliance is recorded but excluded, treating free-form human and structured AI-generated reviews equally.

During generator training the judge runs in parallel with RevUtil (Sadallah et al., 2025)[2], a Llama-3.1-8B-Instruct adapter that scores the whole review. A review passes only when both averages reach 3, and the quality score is the smaller of the two; when RevUtil returns no parsable score, the judge's average is used in its place.

Table 9: Quality-judge rubric. The quality score averages the first four dimensions; format compliance is reported separately and does not enter the score or the eligibility rule.

Dimension	Question put to the judge
Coherence	Does the review make sense as a whole, with clear logical flow and no word salad?
Paper specificity	Is the review grounded in concrete details from this paper rather than generic academic criticism?
Usefulness	Would the authors receive useful guidance from this review?
Non-repetition	Does the review avoid repetitive phrasing, repeated claims, and filler?
Format compliance	Does the review have a recognizable peer-review structure?

Eligibility is decided from the numbers: a review is accepted whenever its mean score is at least 3 and no individual dimension falls below 3, and rejected whenever its mean score is below 3, overriding the returned flag in both directions. This constraint rejects generic or incoherent evasive reviews during generator training and mechanism search.

B.3 Quality-Constrained Group Credit

Let V contain the eligible reviews in a four-review group. Eligibility requires valid output formatting, a quality score of at least 3 on a five-point scale, an accepted length of 180–840 words, and no input-artifact exploitation. When $|V| \geq 2$ and the evasion-score range within V is at least .01, the basic advantage for $i \in V$ is

$$\widetilde{A}_i = E(r_i) - \frac{1}{|V|} \sum_{j \in V} E(r_j). \tag{6}$$

The basic advantage is zero otherwise, and is not divided by a group standard deviation. Truncation, length, and input-artifact failures receive negative credit relative to eligible peers, followed by recentering over the affected samples and eligible peers. Formatting failures are handled by a separate reward term. Other quality failures do not enter the evasion baseline. Groups without an eligible review use only nonpositive length-recovery credit where applicable.

B.4 Policy Optimization

Each update samples 12 paper–rating–strategy conditions with four independent reviews per condition, yielding 48 reviews. We optimize all generator parameters with the clipped GRPO (Shao et al., 2024) objective, using a clipping parameter of .2, one policy epoch per rollout batch, and no KL penalty. AdamW (Loshchilov & Hutter, 2019) uses a learning rate of 3×10^{-6}, weight decay of .01, and gradient clipping at 2. Detector and quality scorers remain frozen throughout the update.

B.5 Generator Evaluation

Domain-specific generators are independently trained from the same Qwen3.5-9B (Qwen Team, 2026) base model. Before and after training, we use the same 17 held-out papers per domain,

[2] https://huggingface.co/boda/RevUtil_Llama-3.1-8B-Instruct_score_only

ratings, prompts, and sampling settings, generating four reviews per condition. ML uses all three strategy conditions; Nature Communications and PeerJ use paper-only generation.

A review is eligible if its output format is valid, the quality judge accepts it as a valid review, and its mean quality-dimension score is at least 3 out of 5. Eligibility sets are fixed before detector evaluation and shared by all thirteen detectors. Before and after training retain their respective eligible sets: 198 and 203 ML reviews, 68 and 50 Nature Communications reviews, and 67 and 50 PeerJ reviews. Evasion is the number of eligible reviews classified as non-AI divided by the number of eligible reviews. Pooling across domains gives denominators of 333 before and 303 after training.

Each detector is calibrated on the same 1,711 human-written PeerShift reviews (545 ML, 583 Nature Communications, and 583 PeerJ); the human reviews of the 51 generator-evaluation papers are excluded, so that no paper contributes both calibration reviews and evaluated reviews. Thresholds are determined from human calibration scores, and tied scores are not split. Tables 10 and 11 report domain-wise evasion at 1% and 5% human FPR, respectively. At 5% FPR, empirical false-positive rates range from 4.44% to 4.97% on this calibration set.

At 1% FPR, evasion against PEERREVEAL-BASE increases by 48.99, 63.65, and 6.96 percentage points on ML, Nature Communications, and PeerJ, respectively; paired-paper bootstrap 95% confidence intervals are [40.20, 56.78], [47.39, 79.41], and [-0.70, 14.25], using 10,000 resamples. At 5% FPR, the corresponding increases are 74.27, 85.06, and 76.57 points, with confidence intervals [68.88, 79.90], [74.14, 94.55], and [63.03, 89.15]. Evasion against EDITLENS (Thai et al., 2026) also increases in every domain at 1% FPR, by 34.26, 16.18, and 30.84 points with intervals [24.64, 44.25], [4.41, 30.88], and [17.65, 44.40]; none of these intervals crosses zero. After generator training, adversarially trained ROBERTA has the lowest pooled evasion at 1% human FPR (42.90%), while regular ROBERTA has the lowest at 5% (30.03%). Before training, EDITLENS is almost never evaded at 5% human FPR: only 1 of the 333 eligible reviews is classified as non-AI, so editing extent alone separates untrained generator output from human writing almost perfectly.

Table 10: Domain-wise evasion rates (%) at 1% human FPR. Each cell shows before → after training. † marks detector families used for generator-training feedback.

Detector	ML	Nature	PeerJ
LOGRANK	100.00 → 100.00	98.53 → 100.00	100.00 → 100.00
FAST-DETECTGPT	100.00 → 100.00	69.12 → 100.00	89.55 → 100.00
BINOCULARS	100.00 → 100.00	100.00 → 100.00	100.00 → 100.00
RADAR†	97.98 → 100.00	100.00 → 100.00	100.00 → 100.00
ANCHOR†	73.74 → 99.51	75.00 → 100.00	85.07 → 100.00
TF	83.84 → 91.13	92.65 → 100.00	97.01 → 100.00
MAGE	98.99 → 100.00	100.00 → 100.00	100.00 → 100.00
EDITLENS	48.99 → 83.25	83.82 → 100.00	67.16 → 98.00
ROBERTA (regular)	3.54 → 54.19	4.41 → 16.00	49.25 → 30.00
ROBERTA (adversarial)	0.51 → 63.05	0.00 → 4.00	2.99 → 0.00
SEM-DETECT (regular)†	95.45 → 100.00	54.41 → 100.00	59.70 → 100.00
PEERREVEAL-BASE†	51.01 → 100.00	32.35 → 96.00	91.04 → 98.00
PEERREVEAL†	2.53 → 89.16	2.94 → 68.00	20.90 → 8.00
Eligible reviews	198 → 203	68 → 50	67 → 50

Table 11: Domain-wise evasion rates (%) at 5% human FPR. Each cell shows before → after training. † marks detector families used for generator-training feedback.

Detector	ML	Nature	PeerJ
LogRank	100.00 → 100.00	98.53 → 100.00	100.00 → 100.00
Fast-DetectGPT	100.00 → 100.00	42.65 → 100.00	55.22 → 96.00
Binoculars	100.00 → 100.00	100.00 → 100.00	100.00 → 100.00
Radar†	84.85 → 73.40	91.18 → 96.00	80.60 → 82.00
Anchor†	41.41 → 92.61	42.65 → 100.00	41.79 → 100.00
Tf	44.95 → 67.00	75.00 → 100.00	80.60 → 94.00
Mage	91.41 → 100.00	98.53 → 100.00	92.54 → 100.00
EditLens	0.51 → 20.20	0.00 → 94.00	0.00 → 70.00
RoBERTa (regular)	3.54 → 37.44	2.94 → 10.00	37.31 → 20.00
RoBERTa (adversarial)	0.51 → 62.56	0.00 → 4.00	2.99 → 0.00
Sem-Detect (regular)†	69.19 → 100.00	50.00 → 100.00	50.75 → 100.00
PeerReveal-Base†	24.75 → 99.01	2.94 → 88.00	13.43 → 90.00
PeerReveal†	0.00 → 63.05	0.00 → 40.00	0.00 → 0.00
Eligible reviews	198 → 203	68 → 50	67 → 50

C Attention-Mask Ablation

Compared structures. Figure 6 shows the 15 visibility patterns. Rows denote query blocks and columns denote key/value blocks. 1 permits attention; 0 blocks it. All four Transformer layers use the same mask. Our mask lets target claims read reference claims and exchange information with textual features, while reference claims attend within their own block. CLS reads every block. Alternatives change these connections or remove an input source while preserving model capacity.

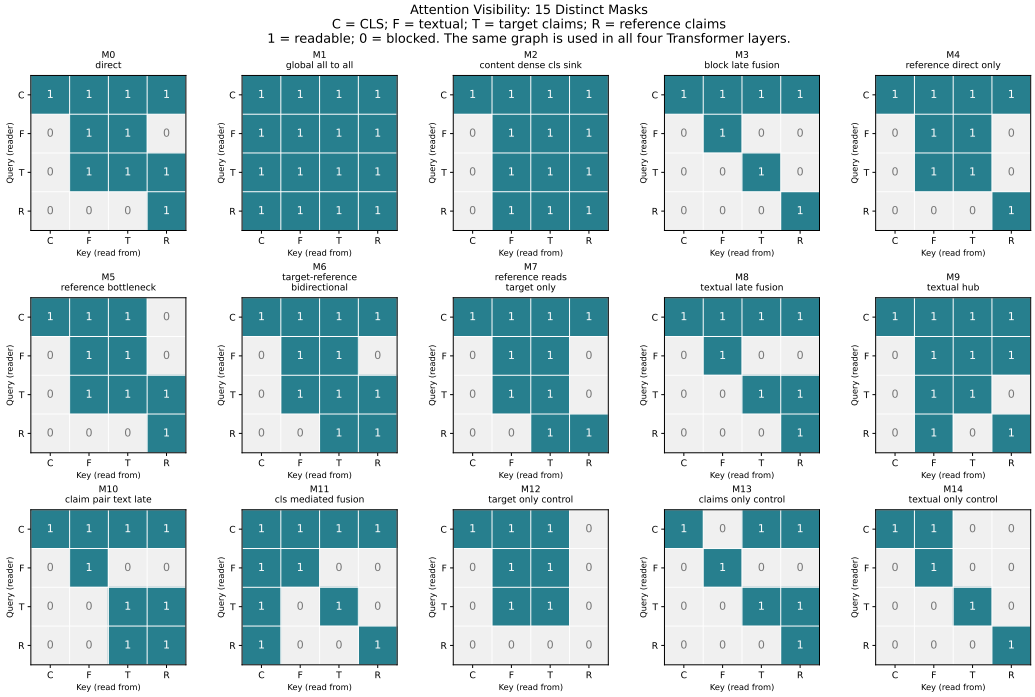


Figure 6: The 15 attention masks. C denotes CLS, F textual features, T target claims, and R reference claims. Our mask appears at the upper left.

Training and selection. Structures use matched initialization, data, sampling, losses, and training budgets. Each undergoes regular and adversarial training, with final model selection on development data. Table 12 gives per-domain and pooled Macro-F1 on PeerShift.

Table 12: Attention-mask results on PeerShift after development-set selection: Macro-F1 (%).

ID	Attention structure	ML	Nature	PeerJ	Pooled
M0	**Our directed attention mask**	**79.95**	**81.98**	85.18	**82.40**
M1	Fully connected attention	78.22	80.00	82.72	80.39
M2	Dense content attention with CLS readout	77.82	78.56	82.24	79.71
M3	Blockwise late fusion	78.25	80.59	84.49	81.17
M4	Reference access through CLS only	79.78	80.18	82.23	80.78
M5	Reference bottleneck	78.30	79.79	84.01	80.80
M6	Bidirectional target–reference attention	78.55	76.62	81.26	78.85
M7	Reference reads target only	79.37	77.79	81.03	79.46
M8	Textual late fusion	77.06	80.51	84.50	80.77
M9	Textual hub	78.76	78.99	84.57	80.79
M10	Claim-pair attention with textual late fusion	76.79	78.75	83.82	79.95
M11	CLS-mediated fusion	78.25	81.37	83.21	81.08
M12	Without reference information	77.74	75.79	82.27	78.66
M13	Without textual information	77.02	80.57	**85.62**	81.12
M14	Textual information only	46.24	45.61	51.64	48.58

We compare the directed blockwise mask with 14 alternatives under matched capacity and training; ours is best. The two evidence sources are complementary: Macro-F1 drops 3.74 points without reference claims and 1.28 without textual features, which alone reach 48.58%.

D Training Data, Protocols, and Additional Results

D.1 Regular and Adversarial Training Corpora

The ML regular corpus follows Sem-Detect. For Nature Communications and PeerJ, we collect manuscript–human-review pairs from public review records and generate corresponding regular AI reviews and faithful rewrites using conventional prompts. The PeerJ corpus is restricted to Psychology or Social Sciences. Paper-level partitions separate training and development from the final evaluation. Table 13 reports the training corpora.

Table 13: Three-domain training corpora. Both stages use regular-corpus human reviews.

		Regular			Adversarial	
Domain	Human	AI	Rewrite	Total	AI	Rewrite
ML	2,088	4,580	8,392	15,060	5,053	1,259
Nature	926	1,600	3,704	6,230	300	413
PeerJ	2,075	4,488	8,299	14,862	960	479
Total	5,089	10,668	20,395	36,152	6,313	2,151

The regular pools contain 541 ML, 400 Nature Communications, and 1,122 PeerJ papers. The ML regular writer pool includes DeepSeek V3.1 (DeepSeek-AI, 2024)[3], Gemini 2.5 Flash (Comanici et al., 2025)[4], Gemini 2.5 Pro[5], and Qwen3-235B (Yang et al., 2025). The new-domain regular pool uses DeepSeek V3.2 (DeepSeek-AI et al., 2025), GPT-oss-120B[6], Mistral Large 3[7], and Qwen3-235B. PeerShift instead applies the frozen reasoning and reconstruction mechanisms in Appendices A.4–A.5, rather than regular training prompts on new papers.

[3] https://huggingface.co/deepseek-ai/DeepSeek-V3.1

[4] https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash

[5] https://ai.google.dev/gemini-api/docs/models/gemini-2.5-pro

[6] https://huggingface.co/openai/gpt-oss-120b

[7] https://huggingface.co/mistralai/Mistral-Large-3-675B-Instruct-2512

D.2 ML-ONLY COMPARISON

Both detectors use the original ML regular training corpus, without new-domain or adversarial data. SEM-DETECT uses its original nine-feature LightGBM (Ke et al., 2017) classifier. PEERREVEAL-BASE uses the factorized classification head, trained on the 85% training partition and selected on the remaining 15% development partition. The same selected checkpoint is evaluated on Original, OOD-M, OOD-M+P, and PeerShift-ML. Table 14 gives the class recalls. Reproduction configurations accompany the code.

Table 14: Class recall (%) for the ML-only models in Table 3.

Detector	Test set	AI	Rewrite	Human
SEM-DETECT	Original	91.55	92.31	63.93
	OOD-M	68.86	82.31	63.69
	OOD-M+P	68.06	76.65	63.69
	PeerShift-ML	15.08	31.97	63.93
PEERREVEAL-BASE	Original	98.76	96.48	92.13
	OOD-M	94.19	94.52	93.11
	OOD-M+P	86.61	88.70	93.30
	PeerShift-ML	30.00	31.31	92.13

D.3 THREE-DOMAIN TRAINING AND MODEL SELECTION

PEERREVEAL-BASE is trained on the three-domain regular corpus; PEERREVEAL continues training on a mixture of regular and adversarial reviews. SEM-DETECT is trained separately on the same corpora while retaining its nine feature definitions and reference reviews. Preprocessing is fitted on training data; model selection uses three-domain mean development Macro-F1.

Development sets use separate papers and the same generation mechanisms as PeerShift, containing 1,101 ML, 207 Nature Communications, and 255 PeerJ reviews. The mechanisms are shared between development and test; they are the frozen artifacts of the search in Appendices A.4–A.5. Final evaluation uses PeerShift, with overall metrics computed from pooled predictions. Final training configurations and execution commands accompany the code.

D.4 BASELINE IMPLEMENTATIONS

Released detectors keep their published scoring models and decision procedures; only the calibration data and the evaluation set change. LOGRANK (Ippolito et al., 2020) scores reviews with GPT-2 Medium[8]. FAST-DETECTGPT (Bao et al., 2024) uses GPT-Neo-2.7B[9] with the analytic sampling-discrepancy estimator. BINOCULARS (Hans et al., 2024) contrasts Falcon-7B as observer with Falcon-7B-Instruct as performer[10] over the first 512 tokens. RADAR keeps its released RADAR-Vicuna-7B[11] weights without fine-tuning and averages scores over 512-token chunks covering the whole review. MAGE (Li et al., 2024) uses its released detector on the whole review. EDITLENS keeps its released RoBERTa-large classifier over four AI-editing-extent buckets together with the authors' text cleaning; we read its AI score as the expected bucket index normalized to $[0, 1]$ and average it over consecutive 510-token chunks so that the whole review is covered. TF (Kumar et al., 2024) scores reviews from part-of-speech-tagged adjective, noun, and adverb frequency profiles estimated on AI and human reviews, and is the only baseline that uses no neural language model.

ANCHOR embeds the target review and one same-paper GPT-5.5 reference review with the same frozen Qwen3-Embedding-0.6B (Zhang et al., 2025) encoder that PEERREVEAL uses, and scores their cosine similarity.

[8] https://huggingface.co/openai-community/gpt2-medium
[9] https://huggingface.co/EleutherAI/gpt-neo-2.7B
[10] https://huggingface.co/tiiuae/falcon-7b
[11] https://huggingface.co/TrustSafeAI/RADAR-Vicuna-7B

Our RoBERTa baseline uses RoBERTa-base with a three-class classification head on review text. Its regular and adversarial versions use the same respective training corpora as PeerReveal-Base and PeerReveal.

D.5 Per-Class Results

Table 15: PeerShift class recall (%) for PeerReveal-Base and PeerReveal (Table 2).

Training	Domain	AI	Rewrite	Human
Base	ML	56.89	30.16	93.77
	Nature	72.00	72.00	76.67
	PeerJ	69.33	87.50	74.67
Full	ML	93.77	62.62	84.43
	Nature	91.67	75.50	78.83
	PeerJ	88.67	85.17	81.33

Rows and columns of the pooled confusion matrices follow AI, rewrite, human:

$$C_{\text{Base}} = \begin{bmatrix} 1195 & 539 & 76 \\ 120 & 1141 & 549 \\ 32 & 298 & 1480 \end{bmatrix}, \qquad C_{\text{Full}} = \begin{bmatrix} 1654 & 145 & 11 \\ 152 & 1346 & 312 \\ 31 & 303 & 1476 \end{bmatrix}. \tag{7}$$

Their Macro-F1 scores are 70.52% and 82.40%.

Tables 16 and 17 break the AI and rewrite recalls down by writer model and by mechanism (Appendix A.4). PeerReveal matches or improves PeerReveal-Base for most writer-model and mechanism groups, with the largest gains where PeerReveal-Base was weakest: Grok 4.3 AI reviews rise from 1.64%, 6.67%, and 18.33% to 86.89%, 80.00%, and 76.67% on ML, Nature Communications, and PeerJ, and ML rewrites by Grok 4.3 and DeepSeek V4 Pro from 14.75% and 14.75% to 61.48% and 65.57%. For AI-generated reviews, recall varies more across writer models than across mechanisms within a domain; the hardest AI writers for PeerReveal are MiniMax M2.5 on ML (77.05%), MiniMax M2.5 on Nature Communications (76.67%), and Gemini 3.1 Pro Preview on PeerJ (70.00%).

Table 16: PeerShift recall (%) of PeerReveal-Base (Base) and PeerReveal (Full) by writer model. Each AI cell covers 61 (ML) or 60 (Nature, PeerJ) reviews and each rewrite cell 122 or 120.

		ML		Nature		PeerJ	
Class	Writer model	Base	Full	Base	Full	Base	Full
AI	Gemini 3.1 Pro Preview	37.70	93.44	31.67	83.33	33.33	70.00
	DeepSeek V4 Pro	75.41	98.36	86.67	100.00	75.00	95.00
	MiniMax M2.5	40.98	77.05	65.00	76.67	73.33	76.67
	Mistral Large 3 (675B)	96.72	100.00	91.67	83.33	88.33	93.33
	Qwen3-Next-80B-A3B	98.36	100.00	98.33	98.33	96.67	100.00
	GLM-5	39.34	95.08	66.67	95.00	48.33	83.33
	Kimi K2.5	70.49	100.00	85.00	100.00	75.00	96.67
	Grok 4.3	1.64	86.89	6.67	80.00	18.33	76.67
	Claude Sonnet 4.6	68.85	91.80	90.00	100.00	86.67	96.67
	GPT-5.5	39.34	95.08	98.33	100.00	98.33	98.33
Rewrite	Gemini 3.5 Flash	59.84	73.77	80.00	84.17	86.67	84.17
	DeepSeek V4 Pro	14.75	65.57	73.33	74.17	85.83	85.00
	Kimi K2.5	50.00	76.23	80.00	72.50	93.33	86.67
	Grok 4.3	14.75	61.48	68.33	79.17	85.00	86.67
	GPT-5.5	11.48	36.07	58.33	67.50	86.67	83.33

D.6 Additional Binary Results

Three-class models contribute $P(\text{AI})$, and ROC curves and AUROC are computed from continuous scores. Binary Macro-F1 uses discrete predictions. Three-class detectors predict AI when $P(\text{AI}) \geq$

Table 17: PeerShift recall (%) of PEERREVEAL-BASE (BASE) and PEERREVEAL (FULL) by mechanism; IDs follow Tables 5 and 6.

AI mechanism	Reviews	BASE	FULL
ML-A1	152	55.26	94.08
ML-A2	152	56.58	96.05
ML-A3	153	56.21	91.50
ML-A4	153	59.48	93.46
NC-A1	150	74.00	90.00
NC-A2	150	73.33	92.67
NC-A3	150	68.67	90.00
NC-A4	150	72.00	94.00
PJ-A1	200	71.50	93.00
PJ-A2	200	67.00	88.00
PJ-A3	200	69.50	85.00

Rewrite mechanism	Reviews	BASE	FULL
ML-R1	102	21.57	58.82
ML-R2	102	40.20	69.61
ML-R3	102	22.55	57.84
ML-R4	102	35.29	62.75
ML-R5	101	27.72	63.37
ML-R6	101	33.66	63.37
NC-R1	100	62.00	67.00
NC-R2	100	62.00	74.00
NC-R3	100	89.00	84.00
NC-R4	100	70.00	75.00
NC-R5	100	82.00	82.00
NC-R6	100	67.00	71.00
PJ-R1	120	93.33	85.83
PJ-R2	120	76.67	75.83
PJ-R3	120	88.33	88.33
PJ-R4	120	88.33	84.17
PJ-R5	120	90.83	91.67

.5, without threshold tuning. A binary Macro-F1 near 40% corresponds to predicting almost no review as AI-generated; regular SEM-DETECT reaches this value on PeerShift (Table 1) because its $P(\mathrm{AI})$ rarely exceeds .5 under this rule, despite an AUROC of 77.87%. Every other method uses one global threshold that maximizes binary Macro-F1 on a calibration panel of 2,665 regular-corpus reviews from 145 papers (48 ML, 50 Nature Communications, and 47 PeerJ; 780 AI-generated, 1,508 AI-rewritten, and 377 human-written), none of which shares a paper with PeerShift; ties are resolved toward the higher threshold. Table 18 lists the resulting thresholds. ROC operating points use both human-written and AI-rewritten reviews as negatives. We report the highest empirical TPR with FPR at or below each stated limit, retaining complete score ties without interpolation. All 14 evaluated configurations cover the same 5,430 reviews (1,810 positives and 3,620 negatives).

Table 18: Binary Macro-F1 decision thresholds, selected on the 2,665-review calibration panel. A review is predicted AI-generated when its score reaches the threshold.

Method	Threshold	Calibration Macro-F1 (%)
LOGRANK	−1.1281	41.55
FAST-DETECTGPT	1.4018	56.70
BINOCULARS	−0.9472	49.89
RADAR	0.6339	57.41
ANCHOR	0.8735	86.61
TF	0.2915	54.04
MAGE	0.9998	41.43
EDITLENS	0.4292	56.63

Table 19 gives the complete per-domain results. Figures 7 and 8 show full and low-FPR ROC curves for PeerShift (pooled) and each domain.

Table 19: Per-domain binary results on PeerShift (%): AUROC / TPR@1% / Macro-F1.

Detector	ML	Nature	PeerJ
LogRank	27.88 / 0.00 / 39.98	35.11 / 0.83 / 39.96	41.39 / 1.50 / 40.16
Fast-DetectGPT	60.74 / 13.11 / 58.45	71.01 / 28.50 / 69.12	63.37 / 16.83 / 62.83
Binoculars	8.97 / 0.00 / 25.61	50.99 / 3.00 / 54.81	44.07 / 1.33 / 49.38
Radar	69.84 / 0.49 / 60.87	60.05 / 0.50 / 53.10	73.40 / 5.50 / 64.16
Anchor	86.17 / 19.51 / 73.72	87.53 / 29.67 / 73.04	88.19 / 30.50 / 75.63
Tf	50.45 / 9.02 / 51.07	61.77 / 7.00 / 52.05	53.76 / 4.67 / 49.81
Mage	56.14 / 0.00 / 39.98	55.97 / 0.17 / 40.00	46.19 / 0.17 / 40.00
EditLens	80.70 / 0.00 / 66.18	77.96 / 0.17 / 67.33	61.02 / 0.17 / 55.86
RoBERTa (reg.)	78.68 / 61.97 / 71.14	77.70 / 50.50 / 66.61	76.48 / 39.67 / 64.98
RoBERTa (adv.)	89.07 / 75.41 / 85.17	86.93 / 71.00 / 81.19	91.22 / 75.17 / 84.81
Sem-Detect (reg.)	71.62 / 31.48 / 40.00	81.16 / 18.00 / 40.00	82.38 / 32.17 / 40.00
Sem-Detect (adv.)	87.78 / 15.57 / 72.96	87.14 / 20.83 / 77.16	86.72 / 22.17 / 73.49
PeerReveal-Base	94.02 / 54.10 / 80.39	91.60 / 27.83 / 83.64	92.84 / 29.83 / 84.12
PeerReveal	98.78 / 77.54 / **94.73**	97.55 / 50.50 / **91.79**	97.61 / 49.83 / **92.51**

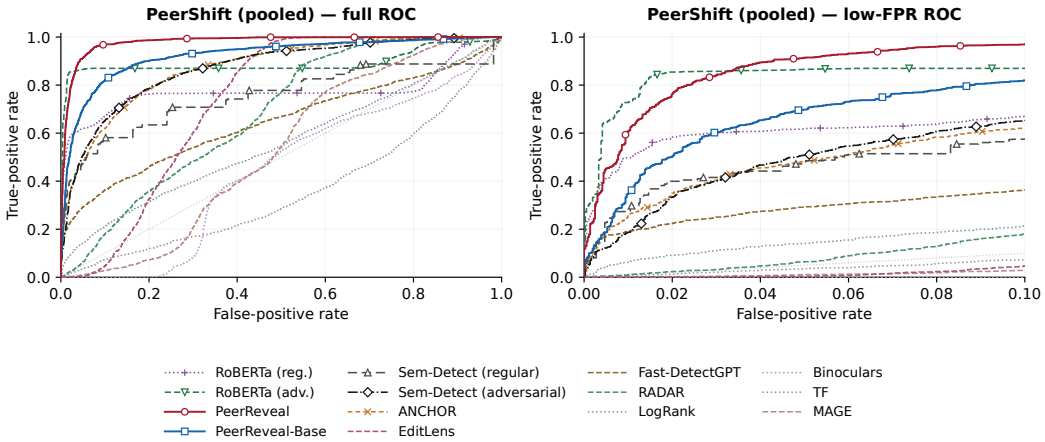


Figure 7: Pooled binary ROC curves on PeerShift. Left: full ROC. Right: FPR from 0 to 0.1.

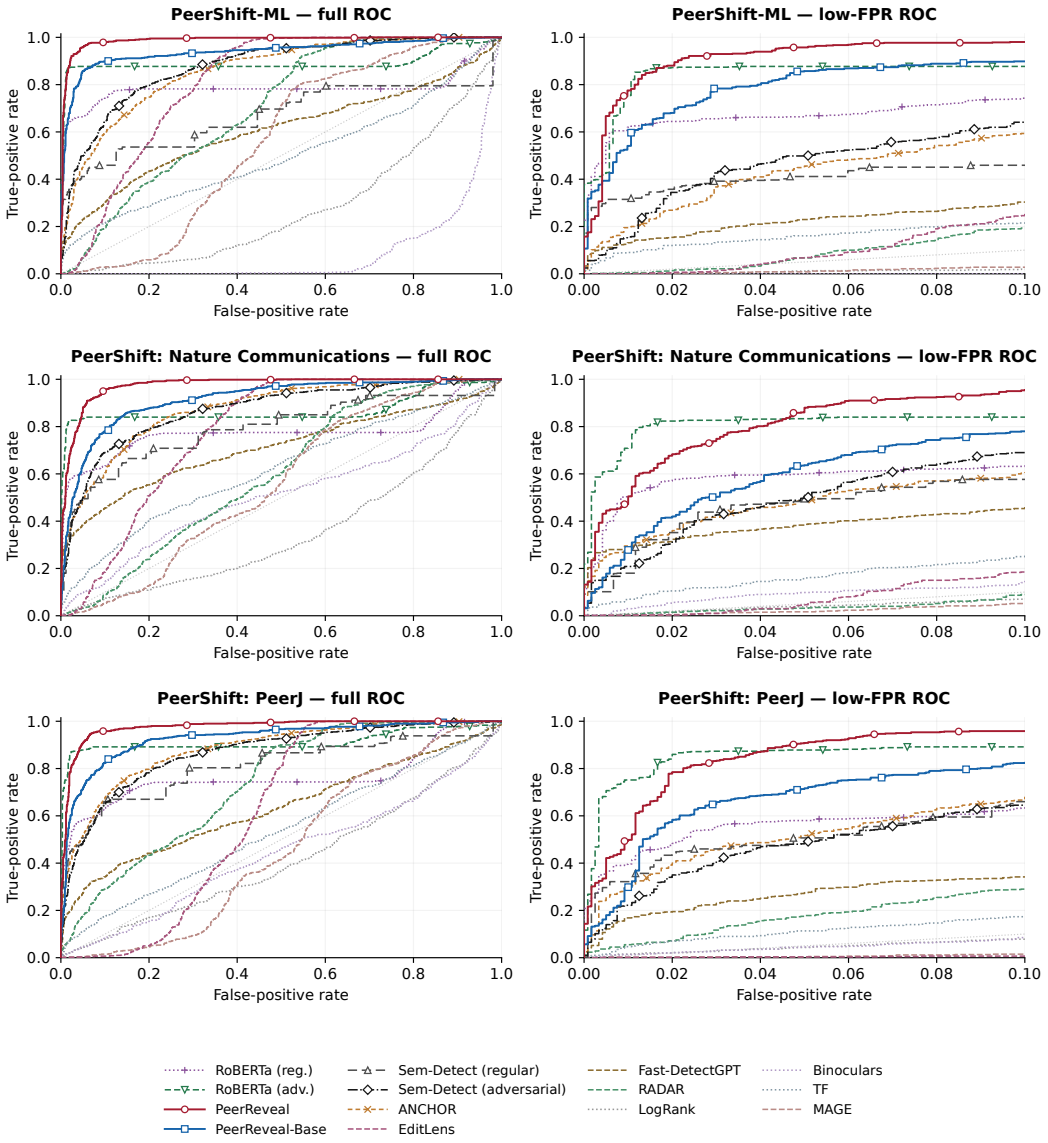


Figure 8: Binary ROC curves on PeerShift for the same detectors evaluated on ML (top), Nature Communications (middle), and PeerJ (bottom). Left: full ROC. Right: FPR from 0 to 0.1.

D.7 HUMAN-VERSUS-AI BINARY RESULTS

Table 20 repeats the binary evaluation with AI-rewritten reviews removed, so that fully AI-generated reviews are compared with human-written reviews only, as in the binary setting of SEM-DETECT. AUROC and TPR at 1% FPR use the same scores as Table 1. For Macro-F1, three-class detectors keep the $P(\mathrm{AI}) \geq .5$ rule, and every other method uses a threshold re-selected by the same rule on the 780 AI-generated and 377 human-written reviews of the calibration panel. PEERREVEAL has the highest AUROC (99.23%) and TPR at 1% FPR (87.90%). EDITLENS achieves 98.04% AUROC and the highest Macro-F1 (96.07%), but falls to 73.11% AUROC with rewrites included as negatives (Table 1).

Table 20: Pooled human-versus-AI binary detection on PeerShift (%). Positive: AI-generated (1,810); negative: human-written (1,810). AI-rewritten reviews are excluded.

Detector	AUROC	TPR@1%	Macro-F1
LOGRANK	32.90	0.66	34.45
FAST-DETECTGPT	66.63	20.17	62.40
BINOCULARS	47.11	0.00	50.34
RADAR	77.56	4.97	66.12
ANCHOR	86.01	21.60	77.25
TF	65.62	9.12	60.43
MAGE	55.22	0.17	56.35
EDITLENS	98.04	37.35	**96.07**
ROBERTA (regular)	75.82	59.12	62.42
ROBERTA (adversarial)	86.97	85.64	80.38
SEM-DETECT (regular)	79.28	36.80	33.33
SEM-DETECT (adversarial)	86.65	14.86	70.85
PEERREVEAL-BASE	95.41	58.84	81.56
PEERREVEAL	**99.23**	**87.90**	94.77

D.8 COST AND COMPUTE

Constructing PeerShift is dominated by review generation rather than by detector training. Generating its 3,620 AI-generated and AI-rewritten reviews through API providers cost about $600, roughly $0.17 per review. Detection is inexpensive. Each target review needs one claim-extraction call, which costs about $0.0026 per review with Gemini 2.5 Flash ($0.0013 with batch calls). The three references are generated once per manuscript and rating and shared by every review of that manuscript at that rating; at public Amazon Bedrock prices they cost about $0.02–$0.04 per manuscript and rating, plus three extraction calls. The claim encoder, the reference language model, and the detector run locally, and all features are computed once and reused by PEERREVEAL-BASE, PEERREVEAL, the SEM-DETECT comparisons, and every training stage.

Generator training is the largest GPU cost. The final generators are reached after 67 (ML), 39 (Nature Communications), and 23 (PeerJ) GRPO updates, each sampling 48 complete reviews and scoring them with four frozen detectors and the frozen quality scorers, for $3{,}216$, $1{,}872$, and $1{,}104$ generated reviews respectively. PEERREVEAL training is negligible by comparison: with $300{,}295$ trainable parameters over precomputed features, Stage A runs 30 epochs of 406 steps and Stage B 20 epochs of 406 steps, both at batch size 64 on a single GPU.

E DETECTOR IMPLEMENTATION DETAILS

This appendix specifies the preprocessing pipeline, architecture, and optimization settings for the three-domain PEERREVEAL models. Claim extraction, reference selection, and claim embedding are fixed across PEERREVEAL-BASE, PEERREVEAL, and SEM-DETECT.

E.1 CLAIM EXTRACTION

Each review is decomposed into functionally labeled claims by an LLM extractor; as in SEM-DETECT, this granularity follows retrieval work showing that the unit of text representation strongly affects downstream performance (Chen et al., 2024; Duarte et al., 2024). Reviews taken from the SEM-DETECT release, namely the ML regular corpus and the human-written reviews of PeerShift-ML, keep the claims released with it, which were extracted with Gemini-2.5-Flash; all other reviews are extracted by Gemini 2.5 Flash. A JSON schema requiring one string array per category is supplied in the prompt text; decoding is unconstrained. The output budget is $4{,}096$ tokens, with temperature 0.1 and thinking disabled. The extractor sees only the review text; it never sees the manuscript, the writer model, or the class label. The system prompt is:

> You are given the full text of a peer review. Your task is to extract and organize the reviewer's comments into bullet points under the following categories:
> 1. **Factual Restatement** – Summaries or descriptions of what the paper does, its methods, contributions, or results.
> 2. **Evaluation** – Judgments of quality, including both strengths (positive evaluations) and weaknesses/limitations (negative evaluations).
> 3. **Constructive Input** – Actionable suggestions or recommendations for improvement.
> 4. **Clarification Dialogue** – Questions directed to the authors or requests for clarification.
> 5. **Meta-Commentary** – Remarks about the broader context, such as fit for the venue, clarity of writing, novelty, or overall recommendation.
> Instructions:
> – For each category, extract bullet points from the review text that match its definition.
> – Keep each bullet point focused on a single coherent idea, but **do not break ideas into fragments**—preserve the full meaning so the point is self-contained and understandable.
> – Maintain the reviewer's original phrasing where possible, lightly compressing or restructuring only for clarity.
> – If the review contains no content for a category, leave it blank.
> – Do not add information that is not present in the review.

Following SEM-DETECT, we retain evaluation, constructive-input, and clarification-dialogue claims and discard factual restatements and meta-commentary. Claims follow the category order above and extractor order within each category.

Table 21 reports mean claim counts per category on the 5,430 PeerShift reviews. AI-generated reviews carry the most evaluation claims (18.3 against 6.9 for human reviews), while human reviews carry the most constructive-input claims (7.2 against 4.5). Rewrites are the shortest in claim terms. Appendix F reproduces one paper's three reviews in full.

Table 21: Mean claims per review by category on PeerShift, with the three retained categories marked. Retained counts are computed before the 128-claim cap.

Class	Factual restatement	Evaluation*	Constructive* input	Clarification* dialogue	Meta commentary	All	Retained
AI	13.3	18.3	4.5	3.6	2.6	42.3	26.4
Rewrite	4.5	8.5	6.0	2.3	1.6	22.9	16.8
Human	6.3	6.9	7.2	4.1	1.5	26.0	18.2

E.2 REFERENCE REVIEWS

Each target review is paired with three AI reviews of the same manuscript, generated in advance with conventional prompts and matched to its requested rating when available. References are selected once, before any training, and the selected triple is frozen: it does not change between PEERREVEAL-BASE and PEERREVEAL, between training stages, or between PEERREVEAL and SEM-DETECT.

For PeerShift-ML, the reference pool comprises Gemini 2.5 Flash, Gemini 2.5 Pro, DeepSeek V3.1, and Qwen3-235B-A22B, with three references selected per target. PeerShift's Nature Communications and PeerJ subsets use one reference each from Qwen3-235B-A22B, DeepSeek V3.2, and Mistral Large 3. These PeerShift references use SEM-DETECT's base prompt: given the manuscript and

requested recommendation, it asks for a concise, critical, constructive review with a paper summary, strengths, weaknesses, questions for the authors, and a final judgement. No searched mechanism is appended. The new-domain training corpora select three references from their regular AI pool, which additionally includes GPT-oss-120B; these references do not condition on a requested rating.

The three references are concatenated into one pooled block in the frozen selection order, using the same retained categories as the target. The pooled block contains 61.8 claims per review on average (median 61, 10th percentile 50, 90th percentile 74, maximum 96), so the 128-position reference cap in Table 22 is never reached on any evaluated review.

E.3 Claim Encoder and Textual Features

Retained claims are encoded by a frozen Qwen3-Embedding-0.6B sentence encoder in half precision with a batch size of 32 and left-side padding, producing one $1{,}024$-dimensional vector per claim. The encoder is never fine-tuned; the only learned transformation of claim content is the shared linear projection into the 64-dimensional model space described in Section 4.1. SEM-DETECT uses the same frozen encoder, so the gap between the two systems isolates how claim relationships are modeled.

The four review-level textual features are computed on the target review only, with Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)[12] in half precision as the reference language model and a maximum length of $4{,}096$ tokens: mean token perplexity, the FAST-DETECTGPT sampling discrepancy estimated from $10{,}000$ samples on the first 800 tokens, the percentage of tokens inside the model's top-200 predictions, and mean predictive entropy. The four values are standardized with training-set statistics that are frozen into the checkpoint, with means $(7.83, -4.47, .978, 1.654)$ and standard deviations $(2.60, 2.21, .011, .270)$ in the order above, and are encoded as feature tokens (Section 4.1).

E.4 Architecture and Optimization

Table 22 lists the architecture, and Table 23 the two training stages. The detector has $300{,}295$ trainable parameters, at least two orders of magnitude fewer than the transformer detectors it is compared against. Both stages use AdamW with a constant learning rate and no warmup, and the optimizer state is reset when Stage B begins. The loss is the sum of a binary cross-entropy on AI versus non-AI and a conditional rewrite-versus-human cross-entropy evaluated on non-AI examples, with unit weight on the conditional term, no class reweighting, and no distillation or parameter-anchoring terms. Model selection maximizes the unweighted mean of the three per-domain development Macro-F1 scores; this selected epoch 10 of Stage A as PEERREVEAL-BASE and epoch 16 of Stage B as PEERREVEAL. Predictions are taken by argmax of the three-class distribution in Equation 3, with no threshold tuning and no ensembling.

[12] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Table 22: PEERREVEAL detector architecture. Attention and MLP dropout are used in both training stages; textual-token and reference-block dropout are enabled only in Stage B.

Component	Setting
Claim encoder	Qwen3-Embedding-0.6B, 1,024-d, frozen
Model width	64
Encoder layers	4
Attention heads	4
Feed-forward width	256
Classification MLP width	128
Attention mask	directed blockwise (Section 4.1)
Attention dropout	.1
MLP dropout	.2
Textual-token dropout	0 (Stage A), .3 (Stage B)
Reference-block dropout	0 (Stage A), .1 (Stage B)
Max target claims	128
Max reference claims	128
Textual features	4
Trainable parameters	300,295

Table 23: Detector training. Stage A produces PEERREVEAL-BASE; Stage B continues from the selected Stage A checkpoint and produces PEERREVEAL.

	Stage A (regular)	Stage B (adversarial)
Initialization	random	Stage A, epoch 10
Optimizer	AdamW, weight decay .01	AdamW, weight decay .05
Learning rate	8×10^{-5}, constant	1×10^{-4}, constant
Batch size	64	64
Steps per epoch	406	406
Epoch budget	30	20
Domain sampling (ML:Nature:PeerJ)	1:1:1	2:1:1
Class sampling (AI:rewrite:human)	1:1:1	1:1:1
Adversarial fraction	0	.75 AI, .5 rewrite
Gradient clipping	1.0	1.0
Selected epoch	10	16

F QUALITATIVE EXAMPLES

This appendix shows one paper from the PeerShift ML evaluation set together with all three of its review classes, so that the same submission, the same subject matter, and the same recommendation are held fixed while only authorship varies. The paper is *Continual Normalization: Rethinking Batch Normalization for Online Continual Learning* (Pham et al., 2022) (ICLR 2022). All three reviews carry a Borderline recommendation. The AI-rewritten review is a reconstruction of the human review shown here. Each review is reproduced in full, exactly as the detectors received it.

Table 24 gives every detector's output on these three reviews. The six three-class detectors are decided by the argmax of their class posterior; the eight binary detectors are decided by the fixed thresholds of Table 18, under which a review is flagged when its score reaches the threshold, and AI-rewritten reviews count as negatives together with human-written ones. Of the 39 decisions made by the thirteen detectors other than PEERREVEAL, twenty are wrong.

Table 24: Detector outputs on the three reviews of *Continual Normalization* (ICLR 2022). Three-class detectors report $P(\mathrm{AI})/P(\mathrm{rewritten})/P(\mathrm{human})$ and decide by argmax; binary detectors report their score and flag a review when it reaches the threshold, treating rewritten and human reviews alike as negatives. Values are rounded to three decimals; * marks a wrong decision.

Detector	Threshold	Human 319 words	AI-rewritten 254 words Kimi K2.5	AI-generated 419 words Grok 4.3
Three-class detectors				
RoBERTa (regular)	argmax	.000/.000/1.000	.000/1.000/.000	.000/1.000/.000*
RoBERTa (adversarial)	argmax	.000/.004/.996	.000/.999/.001	.000/.999/.001*
Sem-Detect (regular)	argmax	.321/.351/.328*	.313/.319/.368*	.315/.319/.367*
Sem-Detect (adversarial)	argmax	.038/.391/.571	.033/.110/.857*	.062/.216/.723*
PeerReveal-Base	argmax	.000/.004/.996	.000/.841/.158	.000/.006/.994*
PeerReveal	argmax	.000/.004/.996	.001/.999/.000	.989/.011/.000
Binary detectors				
LogRank	-1.128	-1.784	-2.266	-2.538^*
Fast-DetectGPT	1.402	0.721	-1.779	-2.686^*
Binoculars	-0.947	-0.214^*	-0.263^*	-0.793
Mage	1.000	0.948	0.000	0.098^*
Radar	0.634	0.888^*	0.884^*	0.737
EditLens	0.429	0.014	0.475^*	0.669
Tf	0.291	0.033	0.078	0.025^*
Anchor	0.873	0.895^*	0.893^*	0.798^*

Human-written review (319 words).

> This paper investigates the problem of cross-task normalization in a non-stationary online continual learning(CL) setting. It argues that batch normalization (BN) is important to CL; however, BN in the current form introduces a bias towards current task, leading to catastrophic forgetting. The paper presents a continual normalization(CN) layer to reduce this bias. In particular, it combines spatial and batch normalization into one layer that is suitable for CL. The authors conducted experiments to evaluate the performance of the proposed method.
>
> Strong Points
>
> The paper takes one of the most import issues in continual learning: non-stationary online CL setting. For me, the problem itself is real and practical.
>
> The proposed approach is reasonable and addresses the problem of cross-task normalization in CL setting.
>
> The proposed adaptive normalization layers can be a plug-in replacement of BN.
>
> The authors made extensive comparison between CN and prior work such as BN, IN, and GN.
>
> Overall, the paper is well written. In particular, the Related Work section has a nice flow and puts the proposed method into context. Despite the method having limited novelty, the method has been well motivated by pointing out the limitations in SOTA methods.
>
> The authors provide code for reproducing the results in the paper.
>
> Weak Points
>
> The proposed adaptive normalization layer is a straight-forward combination of spatial and batch normalization. So the novelty is limited.
>
> Table 2, in DER++ setting, CN's FM score (for catastrophic forgetting) is not as good as GN's. Please explain the reason for that.
>
> Overall, I vote for marginally accepting. I like the idea of cross-task normalization and handling it by the proposed adaptive normalization layer. My major concern is about the limited novelty of the paper and the performance of CN on catastrophic forgetting (see weakness above). Hopefully the authors can address my concern in the rebuttal period.
>
> [After rebuttal]
> After reading the other reviewers' comments and the authors' rebuttal, I confirm my rating.

AI-rewritten review (Kimi K2.5, 254 words).

This paper investigates cross-task normalization in non-stationary online continual learning, arguing that batch normalization introduces a troubling bias toward the current task that ultimately drives catastrophic forgetting. The proposed continual normalization layer attempts to mitigate this by combining spatial and batch normalization, and I find the core idea genuinely appealing—the notion of an adaptive normalization mechanism tailored for continual learning makes practical sense.

The work delivers several strengths worth acknowledging. It addresses a real problem that practitioners face, and CN's plug-in compatibility with existing architectures is a genuine convenience. The comparisons with batch, instance, and group normalization are extensive, the writing—particularly the related work section—is clear, and the provided code supports reproducibility. The motivation underlying the method is sound even if the technical contribution itself feels somewhat familiar.

That said, the novelty remains a sticking point. CN essentially stitches together established normalization techniques in a straightforward manner, which limits its conceptual contribution. More troubling is the empirical result in Table 2 under the DER++ setting, where CN's forgetting measure actually underperforms group normalization. The paper owes readers an explanation for this pattern, as it cuts against the central claim that CN specifically addresses forgetting.

I land at a marginal accept. The cross-task normalization perspective and the proposed adaptive layer are ideas I genuinely like and want to see in the literature. However, the limited novelty combined with the concerning forgetting results temper my enthusiasm—the paper earns its place but not a stronger endorsement. After reviewing the rebuttal and other comments, I maintain this rating.

AI-generated review (Grok 4.3, 419 words).

The paper begins from a concrete and well-motivated observation: on the five-task pMNIST stream, replacing BN's running statistics with oracle moments computed from all training data (denoted BN*) raises final accuracy while leaving learning accuracy essentially unchanged and lowering the forgetting measure (Table 1). The gap is largest without a replay buffer and shrinks once ER supplies a modest amount of past-task data, consistent with the claim that recent-task moments bias normalization of older examples. This toy result directly supports the existence of a cross-task normalization effect under strictly online, single-epoch training.

That observation, however, does not automatically establish the same mechanism as the primary driver of performance differences once the authors move to Split CIFAR-100 and Mini-ImageNet. In Table 2, CN improves both LA and ACC relative to BN under ER and DER++, yet the reduction in FM is frequently comparable to (or smaller than) the reduction already obtained by plain GN. Because GN normalizes entirely within-sample and therefore cannot perform any cross-task moment sharing, the similar or better forgetting numbers under GN already achieve one of the desiderata listed in Section 4.3. The additional gain from CN could therefore be explained either by a successful balance between batch-wise transfer and per-sample adaptivity, or simply by the fact that a two-stage GN-then-BN cascade produces more stable gradients in the low-batch, single-epoch regime irrespective of task ordering. The experiments do not contain a direct measurement—such as the per-task divergence between test-time feature moments under CN versus BN—that would separate these accounts.

The same ambiguity appears in the class-incremental results (Table 4). CN(G=32) yields the highest average accuracy, but the standard deviations remain large at the smallest buffer size and the ranking between CN(G=8) and CN(G=32) is not uniform across memory regimes. Without an ablation that freezes the running statistics or measures how much each method's normalization statistics drift across task boundaries, it remains unclear whether the reported margin is attributable to reduced cross-task bias or to generic optimization advantages of the hybrid normalizer. The long-tailed experiments (Tables 5–6) exhibit still smaller and occasionally reversed gains on minority classes, further indicating that any benefit is sensitive to distribution skew and not uniformly explained by the proposed mechanism.

Consequently, while the initial pMNIST diagnostic is illuminating and CN is a lightweight, drop-in replacement that improves accuracy on the tested benchmarks, the evidence tying those improvements specifically to mitigation of cross-task normalization is indirect and admits plausible alternative interpretations. Stronger causal measurements would be required to move the contribution beyond the borderline.

What the errors look like. The three reviews all discuss the same normalization results and arrive at the same recommendation. PEERREVEAL-BASE assigns the generated and human reviews similarly high human probabilities (.994 and .996): before adversarial training the detector reads a paper-grounded generated review as human with high confidence. PEERREVEAL moves that review to $P(\text{AI}) = .989$ while leaving the human review at $P(\text{human}) = .996$, which is the gain Section 4.2 trains for. The rewritten review separates the systems further still: PEERREVEAL places .999 on the rewritten class, whereas both SEM-DETECT variants call it human, and the eight binary detectors have no way to express the distinction at all, since their only question is whether a review is AI-generated. Among those eight, the errors run in both directions on the same paper — ANCHOR flags the human and rewritten reviews while missing the generated one, and TF, LOGRANK, FAST-DETECTGPT, and MAGE miss the generated review after passing the other two — which is the pattern behind the low binary Macro-F1 values of Table 1.

G ADVERSARIAL GENERATOR EXAMPLE

This appendix shows one review written by the ML generator after adversarial training (Section 4.2). The generator was given *Learning to Guide and to be Guided in the Architect-Builder Problem* (Barde et al., 2022) (ICLR 2022), a Reject recommendation, and the paper-only strategy condition. The quality judge of Appendix B.2 scores the review 5 on all four dimensions, and RevUtil averages 5.0. PEERREVEAL never saw this review during training. The review is reproduced in full, exactly as the detectors received it.

Table 25 gives every detector's output, decided as in Table 24. Ten of the thirteen detectors other than PEERREVEAL miss the review.

Table 25: Detector outputs on the generated review of the Architect-Builder paper (ICLR 2022). Three-class detectors report $P(\text{AI})/P(\text{rewritten})/P(\text{human})$ and decide by argmax; binary detectors report their score and flag the review when it reaches the threshold of Table 18. Values are rounded to three decimals; * marks a missed detection.

Detector	Threshold	AI-generated 317 words Qwen3.5-9B
Three-class detectors		
ROBERTA (regular)	argmax	$1.000/.000/.000$
ROBERTA (adversarial)	argmax	$.999/.000/.000$
SEM-DETECT (regular)	argmax	$.313/.319/.368^*$
SEM-DETECT (adversarial)	argmax	$.156/.006/.839^*$
PEERREVEAL-BASE	argmax	$.001/.003/.997^*$
PEERREVEAL	argmax	$.999/.001/.000$
Binary detectors		
LOGRANK	-1.128	-2.646^*
FAST-DETECTGPT	1.402	-2.503^*
BINOCULARS	-0.947	-1.069^*
MAGE	1.000	0.038^*
RADAR	0.634	0.992
EDITLENS	0.429	0.368^*
TF	0.291	0.011^*
ANCHOR	0.873	0.808^*

Generated review (Qwen3.5-9B, 317 words).

> The paper proposes the Architect–Builder Problem (ABP) and ABIG, an algorithm that learns guiding communication without rewards or demonstrations by alternating between modelling and guiding frames, using shared intent and self-imitation. On a tiny gridworld with up to six blocks, ABIG achieves decent task success and shows some transfer when trained on "place". The idea is intriguing and the analysis of forget/reinforce dynamics is thoughtful. However, the empirical story feels weak: environment is trivial, baselines are sparse (random actions; an oracle ABIG-no-intent that cheats at test), no ablation of priors

> beyond one switch, no comparison to standard MARL/teacher-student methods, and scalability/generalization is barely explored. Writing is dense and sometimes opaque; several claims rely on "interesting" analogies to naming games rather than solid evidence. Overall, interesting setting but not enough substance for acceptance now.
>
> Strengths
>
> Clear formalisation of ABP; nice intuition about stationarity via frames; good analytic insight into how self-imitation reinforces message–action associations and forgets unused ones; clean experiments showing transfer from simple to harder tasks when trained on "place"; clean plots and stats.
>
> Weaknesses
>
> Very narrow evaluation: tiny gridworld, few tasks, no vision/perception, no locomotion beyond navigation/grasp-drop. Baselines are weak and somewhat unfair; no strong MARL/IL/HRI comparators. Ablations lack depth (no testing of frame length, network size, vocab size effects beyond one sweep). Claims about general communication emergence are overstated given the toy domain. Writing is occasionally unclear; some derivations feel hand-waved. No discussion of failure modes or sample efficiency. The "transfer" gains may simply reflect task decomposition rather than genuine protocol emergence.
>
> Questions for Authors
>
> Why no MARL/teacher-student baselines? Can ABIG match them when rewards/ demonstrations are available? What happens with larger grids, more agents, or continuous actions/perception? How sensitive are results to frame design/network/architecture? Does transfer generalize to truly new semantics or just similar geometries? Are claims about naming-game parallels supported empirically? Please clarify oracle baseline at test and justify its fairness.
>
> Final Judgement
>
> Reject

What the detectors see. PEERREVEAL-BASE reads the review as human ($P(\text{human}) = .997$), and so do both SEM-DETECT variants; PEERREVEAL, trained on the adversarial reviews of Section 4.2, assigns $P(\text{AI}) = .999$. ANCHOR compares the review with a same-paper GPT-5.5 reference and misses it (0.808 against 0.873), and TF and the other binary detectors except RADAR all score it on the human side of their thresholds. Both ROBERTA variants also detect the review. RADAR flags it, but also flags both non-AI reviews in Appendix F (0.888 and 0.884 against a threshold of 0.634).

H SCOPE AND LIMITATIONS

PeerShift expands model, workflow, and domain coverage, but cannot exhaust future review-production practices. Performance depends on the quality of claim extraction and manuscript-conditioned references. In particular, distinguishing faithful rewriting from human-authored expression remains harder than recognizing fully AI-generated judgement. Detection should therefore support review-process analysis rather than serve as sole evidence for individual misconduct.