INVERSE-SCALING SIGNATURE IN AGENTIC CINEMATIC SUBTEXT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic pipelines that combine retrieval, decomposition and critic-based refinement are widely assumed to improve as inference-time computation increases. On cinematic subtext detection, which requires pragmatic inference about what characters mean but do not say, the assumption fails. Using LitSubText, 2,606 dialogue instances from 132 film screenplays annotated with 8 intent and 9 theme classes, a single forward pass outperforms every agentic configuration tested. Holding the prompt and the tools fixed and varying only whether the ReAct loop runs, theme macro-F1 falls from 27.3 to 13.0 percent and theme accuracy from 34.2 to 11.7 percent, a significant gap under a screenplay-level cluster bootstrap. The loop breaks 89.0 percent of what the prompt-only configuration answered correctly while repairing 12.0 percent of what it got wrong, so it substitutes predictors rather than partly improving one. Three restraint mechanisms were added to test whether the loss can be recovered, namely Confidence-Gated Retrieval, a Bidirectional Critic that can recommend simplification as well as deepening, and an Interpretation Depth Controller that stops early on critic direction signals. They do not recover it. Instrumented trajectories show that the simplify direction, the only signal expressing restraint, fired zero times across 240 test instances and that the depth controller produced longer trajectories than the unrestrained loop, so two of the three never operated as specified. Probes that hold the input fixed and vary only the candidate locate the reason, because the critic separates wrong candidates from right ones far more sharply than over-detailed ones from concise ones. What it supplies is largely a correctness signal rather than a restraint signal.

1 INTRODUCTION

Cinematic dialogue operates on two levels, the surface exchange of information and an underlying negotiation of power, emotion and unspoken meaning (Grice, 1975), so detecting subtext computationally requires pragmatic inference beyond sentiment, drawing on character relationships, narrative context and dramatic convention.

LLM-based agents perform strongly on complex reasoning through retrieval-augmented generation, chain-of-thought decomposition and iterative self-refinement (Yao et al., 2023; Wang et al., 2023; Madaan et al., 2023), which suggests cinematic subtext detection should also benefit from scaling inference-time computation. It does not. Across eight configurations of increasing agentic complexity on LitSubText, 2,606 annotated dialogue instances from 132 film screenplays, performance peaks with prompt-only inference and then *declines* as agentic machinery is added, with the loop at a fixed prompt taking theme macro-F1 from 27.3 to 13.0% and theme accuracy from 34.2 to 11.7%. This pattern is termed *agentic overthinking*, an inverse-scaling phenomenon (McKenzie et al., 2023; Gema et al., 2025) at inference time.

Two properties of the task make the measurement delicate. The axes are severely imbalanced, the modal class holding 26.3% of test instances on both, so macro-F1 is the headline metric, and the reference labels are machine-generated and reproduce at $\kappa \approx 0.45$ under an independent annotator, so conclusions rest on orderings between systems scored against the same labels.

Three mechanisms for the collapse are stated as testable claims in section 3.2, namely critic confirmation bias, retrieval-induced over-interpretation and category attractiveness bias, under which

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.

the null class is almost never selected. RA-NarrAgent addresses them with **Confidence-Gated Retrieval (CGR)**, gating retrieval on a cosine-similarity threshold, a **Bidirectional Critic (BiCritic)**, extending a reward model with a three-way direction head over DEEPEN, MAINTAIN and SIMPLIFY, and an **Interpretation Depth Controller (IDC)**, turning direction signals into loop control alongside a Null-First Prompt.

The mechanisms do not recover the loss, and why is the more informative outcome. Instrumented trajectories show SIMPLIFY never fired on the test set and the depth controller lengthened trajectories rather than shortening them, so two of the three were not exercised as specified, which a result table cannot distinguish from their being ineffective. Two probes isolating the critic from confounds an ablation cannot separate are contributed alongside this analysis.

2 Related Work

Agent reasoning and self-correction. LLM-based agents combine reasoning, tool use and iterative refinement (Yao et al., 2023; Wang et al., 2024; Schick et al., 2023; Shinn et al., 2023; Madaan et al., 2023; Zhou et al., 2024), but Huang et al. (2024) show they cannot reliably self-correct without external feedback, Kamoi et al. (2024) that self-correction degrades performance where the initial answer was already reasonable, and Stechly et al. (2024) find the same for planning. These results extend that line, since the failure mode is active *degradation* rather than stagnation as reasoning depth increases.

Overthinking and inference-time compute scaling. Test-time compute scaling has drawn attention following reasoning-focused models (OpenAI, 2024; DeepSeek-AI et al., 2025). Chen et al. (2024) identify overthinking in mathematical reasoning, Gema et al. (2025) document inverse scaling across benchmarks and Su et al. (2025) study reasoning length against correctness, while comparisons of scaling strategies (Balachandran et al., 2025; Agarwal et al., 2025) and a vision-language distractor analysis (Bae et al., 2025) find the benefits of extra inference compute task-dependent rather than uniform. The overthinking measured here concerns *interaction effects* between agentic components rather than chain-of-thought length.

Subtext, pragmatics and narrative NLP. Computational pragmatics has studied implicature (Grice, 1975; Jeretic et al., 2020; Ruis et al., 2023), sarcasm (Joshi et al., 2017; Van Hee et al., 2018), theory of mind (Kim et al., 2023), communicating with subtext (Ahuja et al., 2026) and implication in NLI (Havaldar et al., 2025), while narrative NLP addresses story comprehension (Mostafazadeh et al., 2016; Chaturvedi et al., 2017), narrative QA (Kočiský et al., 2018), summarization (Gorinski & Lapata, 2015; Chen et al., 2022; Kryściński et al., 2022), turning points (Papalampidi et al., 2019), character understanding (Brahman et al., 2021; Sang et al., 2022; Iyyer et al., 2016) and narrative-theory-driven methods (Liu et al., 2026). Retrieval-augmented approaches include REALM (Guu et al., 2020), Self-RAG (Asai et al., 2024) and CRAG (Yan et al., 2024), while Shi et al. (2023) show that irrelevant retrieved context degrades performance. LitSubText appears to be the first resource targeting cinematic subtext detection at scale.

3 Background: The Overthinking Problem

3.1 The LitSubText Benchmark

LitSubText comprises 2,606 dialogue instances from 132 professionally written film screenplays across multiple genres and decades, each typically 3–15 lines of character speech with its surrounding scene description. Instances are annotated along two axes, an **intent** axis of 8 classes covering the speaker's communicative strategy and a **theme** axis of 9 covering the dominant thematic undercurrent, each carrying an explicit null class, No Strong Subtext (NS) and No Dominant Theme (NT). Appendix A defines every class and Figure 3 uses the class codes.

The null classes need care. NS does not assert that subtext is objectively absent, only that *within the visible context* the evidence for a specific non-literal reading is insufficient, and NT likewise means no single theme dominates, so the task is to limit assertions to what the evidence supports. The two nulls are independent by definition but not in the data, co-occurring on 518 of 521 instances.

The dataset is split at the screenplay level and the test set is 240 instances from 12 screenplays. Labels are machine-generated, one LLM pass per instance with no human gold label, and an independent annotator of a different model family reproduces them on a 200-instance subset at Cohen's $\kappa = 0.474$ on intent and 0.444 on theme, moderate reliability that bounds every accuracy below (Appendix B).

3.2 The Overthinking Phenomenon

Eight configurations of increasing agentic complexity are evaluated on the same backbone and the same 240 test instances (section 5.1). The shape of the result is stated here because the method was designed against it. Adding context and structured reasoning helps, as expected. Introducing the agent loop *hurts*, on both axes and by a margin larger than any gap within either family, and more deliberation does not recover the loss. The configurations are not strictly nested, since each step changes several variables at once, so the degradation is *configuration-dependent* rather than a smooth function of inference compute (Appendix C.2). Three mechanisms are hypothesized below, each evaluated in section 5. **(H1) Critic confirmation bias.** The critic preferentially endorses elaborate interpretations regardless of grounding, creating a positive feedback loop (Wan et al., 2025). **(H2) Retrieval-induced over-interpretation.** Retrieved dramatic context from elsewhere in the screenplay is imported into literal passages, amplifying the known degradation from irrelevant retrieval (Shi et al., 2023). **(H3) Category attractiveness bias.** Analytically rich labels attract disproportionate predictions at the null class's expense, and the bias precedes any agent machinery, since zero-shot recall on NS is 0.0% across 49 null instances while CI reaches 54.0%. Whether deliberation corrects this floor or deepens it is what the agent configurations answer (section 6).

4 Method: Restraint-Aware NarrAgent

4.1 Overview

RA-NarrAgent retains NarrAgent's ReAct loop over RETRIEVE, ANALYZE and VERIFY and adds one mechanism per hypothesis of section 3.2, gating RETRIEVE with Confidence-Gated Retrieval (section 4.2), replacing the verification critic with a Bidirectional Critic (section 4.3) and wrapping loop control with an Interpretation Depth Controller (section 4.4), as Figure 1 shows.

4.2 Confidence-Gated Retrieval

Standard RAG pipelines return top-k passages regardless of relevance, triggering over-interpretation (H2) since the agent treats context as informative even when no relevant passage exists. CGR adds a hard gate, returning the top-k passages only when $\max_i \text{sim}(\mathbf{q}, \mathbf{p}_i) \geq \theta$ and the empty set otherwise, for query embedding $\mathbf{q}$, passage embeddings $\{\mathbf{p}_i\}$ and a fixed threshold θ. On the empty set the agent receives "No relevant passages found" and falls back on local context. The threshold is $\theta = 0.55$, set from the observed similarity distribution rather than tuned for accuracy, and what the gate withholds is measured on the test trajectories, since no configuration differs from RA-NarrAgent in the gate alone (Figure 4, Appendix C.4).

4.3 Bidirectional Critic

A unidirectional critic produces only a scalar quality score, implicitly rewarding elaboration and creating confirmation bias (H1). BiCritic builds on Qwen2.5-3B (Yang et al., 2024) with LoRA and two heads sharing the backbone, a scalar reward head $r_\phi(x)$ and a three-way *direction head* $f_\phi(x)$ over DEEPEN, MAINTAIN and SIMPLIFY, so it can ask for less interpretation rather than only score what it is given.

Training data. Five tiers are used, **A** (gold), **B** (correct but shallow), **C** (wrong label), **D** (literal paraphrase) and **E** (over-interpretation). Tier E is built programmatically on 959 of 2,126 training instances by replacing the gold label with an interpretively attractive one and instantiating the explanation from four fixed templates asserting unlicensed content. Its rejections are labelled SIMPLIFY, so the head learns restraint as a property of over-asserted claims rather than a reflex toward the null

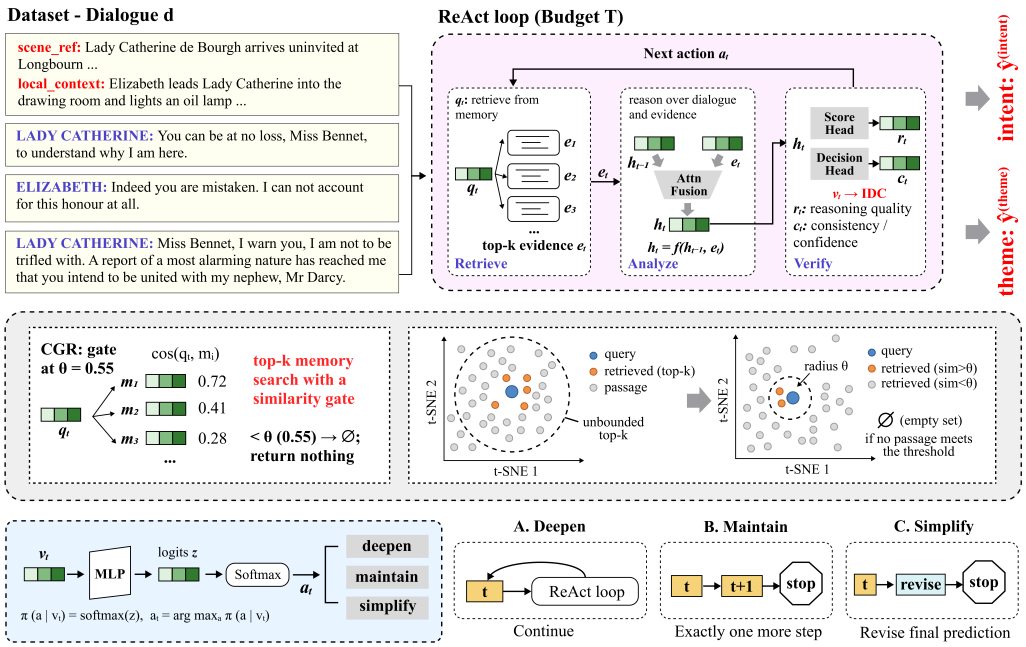


Figure 1: Overview of RA-NarrAgent. A dialogue passage enters the ReAct loop, where Confidence-Gated Retrieval admits a retrieved passage only when its maximum similarity clears θ, the Bidirectional Critic returns a signed direction rather than a scalar score, and the Interpretation Depth Controller converts that direction into a decision about whether to continue or conclude. The three address retrieving irrelevant context, scoring depth without signing it, and continuing past the point where evidence has stopped accumulating. section 5.2 reports that two of the three did not operate as specified on the test set.

classes, and because fixed templates make Tier E narrower than what an agent produces, §5.4 tests whether the signal transfers to model-generated candidates.

Loss function. The reward head uses a Bradley-Terry loss over per-instance pairs with instance-conditional margins. On a null instance a literal paraphrase (D) is nearly right, so its margin against gold drops to 0.15 while a confident wrong reading (C) sits at 0.75, inverting the non-null case where $m(A, D) = 1.0$ against $m(A, C) = 0.5$. Pairs the construction does not license are never formed, so the reward loss is $\mathcal{L}_{\text{BT}} = -\mathbb{E}_{(x_w, x_l)}[\log \sigma(r_\phi(x_w) - r_\phi(x_l) - m(w, l))]$ over the licensed pairs. The direction head is trained with cross-entropy on labels encoding *how the claim set should change* rather than whether the label is right, so asserting too much maps to SIMPLIFY, stopping too early to DEEPEN, and no tier carrying a wrong label maps to MAINTAIN. The combined loss is $\mathcal{L} = \mathcal{L}_{\text{BT}} + \lambda \mathcal{L}_{\text{dir}}$ with $\lambda = 0.3$, and Appendix C gives the tier counts, the margins and the full direction mapping.

The direction head emits a pair $(axis, dir)$, so a SIMPLIFY signal names the axis whose claims are inflated and leaves the other untouched. This per-axis machinery is untested here, since the two nulls co-occur on 518 of 521 instances and no pair separates them.

4.4 Interpretation Depth Controller

The IDC translates BiCritic's direction signal into loop control, addressing category attractiveness bias (H3). A SIMPLIFY signal triggers one revision pass deleting the flagged claims and reconsidering the null class on that axis, then ends the loop, and the answer is formatted from the revised interpretation alone so deleted claims cannot reappear. MAINTAIN grants $T_m = 1$ further step and stops, DEEPEN continues up to the budget T, and every LLM call including the revision pass is charged against it. The heads can disagree, so when $r_\phi < \tau_r$ under MAINTAIN the IDC issues one reconsideration step in which the agent reselects the label from its existing claims, adding none,

which keeps depth and correctness on separate signals. The IDC also injects a *Null-First Prompt* (NFP) requiring the agent to assume no subtext exists unless at least two independent pieces of evidence support a non-null label, shifting the default from "find the subtext" to "verify that subtext exists" (Appendix C.1).

4.5 Training Procedure

Training proceeds in three stages. The **retriever** is a dense passage encoder fine-tuned contrastively on screenplay-level pairs, with same-scene or same-character passages as positives and other screenplays as hard negatives, producing the embeddings CGR uses. **BiCritic** trains the Qwen2.5-3B backbone with LoRA (rank 16, $\alpha = 32$) on the five-tier data of section 4.3, both heads jointly for 3 epochs at learning rate 2×10^{-5} and batch size 32. **Agent SFT** fine-tunes the controller on ReAct trajectories synthesized from the gold labels by a fixed two-tool template, which carries no variation in tool use or stopping behaviour, so the restraint policy comes from the inference-time components (Appendix C).

5 Experiments

5.1 Setup

Metrics. Macro-F1 is the headline metric on both axes, with accuracy alongside it and a constant modal-class predictor as the reference at 26.3% accuracy and 5.2 / 4.6% macro-F1, since under this skew accuracy measures how closely a system matches the label prior while macro-F1 measures discrimination. Parse failures count as errors. Accuracy is agreement with *local* reference labels, so a departure from local gold is ambiguous between unlicensed assertion and legitimate use of remote retrieved evidence, and separating the two would need a licensing-annotated subset, which was not collected. Over-interpretation claims therefore rest on the probes of section 5.4–5.4.

Configurations. Eight configurations share one backbone, Qwen2.5-7B-Instruct, one instance set, one prompt family and the trained retriever. Five are prompt-only, four adding context incrementally and L3+NFP adding the null-first prompt to the last of them, and four run the ReAct loop (Table 1, Appendix C.2). NFP is present in every loop configuration except NarrAgent-Short, so L3+NFP rather than a bare agent isolates the loop at a fixed prompt. ReAct (Yao et al., 2023), Reflexion (Shinn et al., 2023) and Self-RAG (Asai et al., 2024) are not comparison points, since NarrAgent is itself a ReAct loop over task-specific tools and the claim requires comparing one backbone with and without the loop. Each configuration is evaluated once at seed 42, so conclusions are drawn from large orderings rather than small gaps (Appendices C, H).

5.2 Main Results

Table 1 reports a negative finding. On both axes the best prompt-only configuration beats every agent configuration, 18.2 against 15.5% on intent and 27.3 against 13.0% on theme, and additional deliberation does not help. Systems are ordered by macro-F1 because all sit near or below modal-class accuracy, so accuracy cannot separate them while macro-F1 shows they carry label information and still lose to a single forward pass.

RA-NarrAgent has the highest intent macro-F1 of any agent configuration, 15.5% against 8.8 to 13.4%, an ordering no test supports, the difference against NarrAgent+NFP being $+2.1$ points, $[-3.9, +7.5]$, $p = 0.42$. The comparison the tests resolve runs the other way, since on theme RA-NarrAgent is *worse* than NarrAgent+NFP by 7.3 points, $[-12.8, -2.0]$, $p = 0.006$, and of eight confirmatory tests the only one surviving Holm correction is RA-NarrAgent against L3 on theme accuracy, which RA-NarrAgent loses by 27.5 points at $p_{\text{Holm}} = 5.3 \times 10^{-12}$. The restraint mechanisms therefore reorder the agent configurations on intent without separating them and make theme worse, which section 6 traces to the null-class collapse NFP induces on its own.

Theme-axis cost of the loop at a fixed prompt. NarrAgent+NFP against L3 differs only in whether the ReAct loop runs. Theme accuracy falls from 34.2 to 11.7% and macro-F1 from 27.3 to 13.0%, with $\chi^2 = 30.5$, $p < 10^{-7}$, and a cluster-bootstrap accuracy gap of -22.5 points,

Table 1: Main results on the LitSubText test set (240 instances). Both axes are severely imbalanced (modal class 26.3%), so macro-F1 is the headline metric and accuracy differences against the *Majority class* row are not evidence of skill. **Bal.** is balanced accuracy, the mean per-class recall. **Mode** is the share of the 240 predictions taken by the system's own most frequent label, measuring collapse directly. **Parse** is the fraction of labels not resolving to a code, counted as errors. All values %, best per column in bold.

	Intent (8 classes)				**Theme** (9 classes)				
Method	**macro-F1↑**	**Acc.**	**Bal.**	**Mode↓**	**macro-F1↑**	**Acc.**	**Bal.**	**Mode↓**	**Parse↓**
Majority class	5.2	26.3	12.5	100.0	4.6	26.3	11.1	100.0	n/a
Prompt-only configurations (single forward pass, no loop)									
L0 (zero-shot)	17.9	23.3	22.9	49.2	14.1	26.7	20.0	58.3	2.5
L0 + local context	18.0	27.1	23.7	54.2	18.0	30.4	25.2	64.6	4.2
L0 + retrieval	**20.0**	30.0	**27.6**	60.0	18.2	30.8	25.9	62.1	1.7
L0 + narrative context	18.2	22.5	24.0	**33.8**	**27.3**	**34.2**	**31.7**	**33.8**	1.2
L3 + NFP (no loop)	11.5	25.4	16.9	40.8	4.6	5.8	8.5	80.8	**0.4**
Agent configurations (ReAct loop)									
NarrAgent + NFP ($T = 6$)	13.4	18.8	14.7	42.5	13.0	11.7	15.5	52.5	3.8
NarrAgent-Short ($T = 3$)	11.7	15.8	15.8	32.1	5.0	8.3	7.3	36.7	3.3
NarrAgent + Scalar (no direction)	9.6	17.9	12.6	38.3	7.2	8.3	10.5	52.5	5.8
Restraint-augmented agent									
RA-NarrAgent	15.5	20.4	17.2	42.5	5.7	6.7	9.3	61.7	6.2

$[-29.2, -15.8]$, so the effect survives resampling the 12 screenplays, while on intent the same comparison is -3.8 points and no test resolves it. The loop breaks 89.0% of what L3 answered correctly while repairing 12.0% of what it got wrong, and Figure 2b shows this holds for every loop configuration on both axes, at 72.2 to 91.5% broken against at most 21.0% repaired, so the loop substitutes one predictor for another rather than improving on it (Appendix H).

Prior shift under the null-first prompt. NFP raises NS recall from 4.1% under L3 to 79.6% while recall on PT, SI, PP, RG and AD falls to exactly 0.0%, intent macro-F1 drops from 18.2 to 11.5% and theme macro-F1 lands at 4.6%, matching the modal-class predictor to the decimal. The prompt moves the decision threshold until almost everything is called null rather than teaching the model when a reading is unlicensed, so any restraint RA-NarrAgent appears to show has to be separated from this prior shift.

Mechanism-level deviations from the design. A result table cannot distinguish a mechanism that fired and failed from one that never fired, so the trajectories were instrumented directly. Across the 240 test instances RA-NarrAgent issued 487 VERIFY actions, from which the critic returned a parsed direction 70 times, 18 DEEPEN and 52 MAINTAIN, and SIMPLIFY, the only direction expressing restraint, *zero* times against five for NarrAgent+NFP, so the restraint-augmented system emitted fewer restraint signals than the configuration it was meant to improve on, and Table 1 does not evaluate the BiCritic as a restraint mechanism at all. The IDC shows the opposite of its intended effect rather than an absent one, reaching the six-step ceiling on 59 of 240 instances against 41 for NarrAgent+NFP, at mean trajectory lengths of 4.02 against 3.92 steps, since a controller specified to stop early consumes a direction signal that is never SIMPLIFY. CGR is the one mechanism matching its specification, refusing to inject a passage on 39 of 240 instances. An absent restraint effect is therefore equally consistent with the mechanisms being wrong and with two never having engaged.

The two distribution columns show the same effect without reference to the gold labels. Balanced accuracy orders the systems as macro-F1 does, and on theme the best configuration spends 33.8% of its predictions on one label against 36.7 to 61.7% for the loops, RA-NarrAgent highest at 61.7%, so the collapse is visible in the predictions alone.

Read against cost, the ordering inverts the usual one (Figure 2a). The loops consume 3,521 to 6,490 controller tokens per instance against 532 to 1,157 for the prompt-only configurations, and score lower on both axes, by 5.3 and 9.3 points of macro-F1 at the family means.

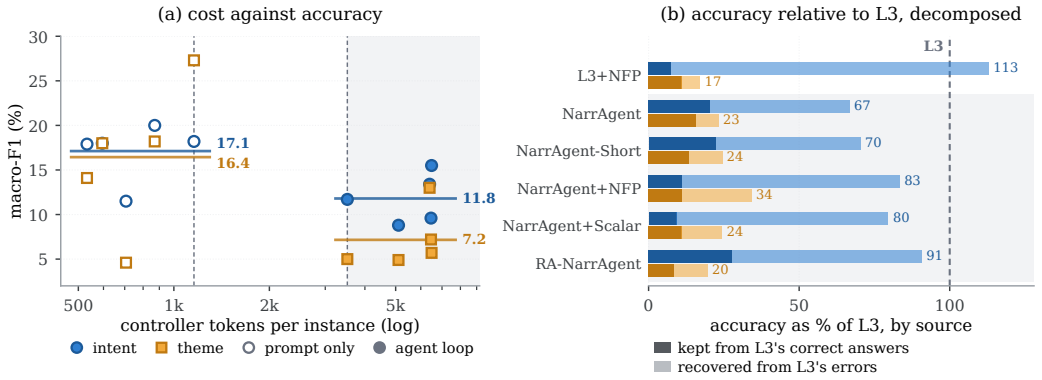


Figure 2: The central result. **(a)** macro-F1 on both label axes against controller tokens per instance, log scale, all ten configurations of Table 1. Markers are hollow for prompt-only and filled for the agent loop, the dashed verticals bound the cost gap from L3 at 1,157 tokens to NarrAgent-Short at 3,521, and the rules are the family means, labelled at their right ends. **(b)** the same accuracy relative to L3 and split by source, using Eq. 1 divided through by L3's accuracy, so 100 is L3 on both axes. Dark is the share of L3's correct answers kept and pale the share recovered from its errors, blue intent and orange theme, shaded rows the agent loop. Normalizing makes the axes comparable, theme having the higher reference, 34.2 against 22.5. The loops reach 67 to 91% of L3 on intent and 20 to 34% on theme, and dark is short everywhere, at most 27.8%, so accuracy is rebuilt from L3's errors rather than inherited.

Table 2: Component ablations on the LitSubText test set (240 instances). The reference is NarrAgent at a 4-step budget, since the arms are leave-one-out from it and share its prompt, budget and SFT adapter. The last two rows are separate runs on base weights at a 6-step budget, *fixed pipeline* calling all five tools in a fixed order and *zero-shot agent* keeping the loop without the adapter. Columns are as in Table 1. Every arm that improves macro-F1 also raises balanced accuracy, so the gains come from covering classes rather than tracking frequent ones. Definitions in Appendix C.2. All values %.

	Intent				Theme			
Configuration	**F1↑**	**Acc.**	**Bal.**	**Mode↓**	**F1↑**	**Acc.**	**Bal.**	**Mode↓**
NarrAgent (reference, $T = 4$)	8.8	15.0	14.0	35.4	4.9	7.9	7.0	51.2
− Critic (no reward, no direction)	10.6	15.0	14.9	38.8	6.7	7.1	11.3	60.4
− Retriever (local context only)	12.9	15.4	17.2	37.1	2.6	4.2	7.8	64.2
− Controller SFT (base weights)	12.4	16.2	15.9	36.7	7.3	10.4	9.9	44.2
Fixed pipeline (no loop)	**15.0**	**16.7**	**21.0**	**41.7**	**13.6**	**19.2**	**19.0**	**31.7**
Zero-shot agent (loop, no SFT)	12.2	15.8	14.5	34.2	4.0	7.5	5.1	41.7

5.3 Ablation Study

Table 2 removes one component at a time from the unrestrained loop and replaces the controller with two degraded variants. Leave-one-out is used rather than a factorial design because the IDC consumes BiCritic's direction signal, so a cell with IDC on and BiCritic off is not a configuration the system can occupy. Each arm differs from the reference row in one component while differing from RA-NarrAgent in five, so the table does not decompose the restraint-augmented system.

Every removal improves on the reference on at least one axis and two of three on both. Dropping the critic raises intent macro-F1 from 8.8 to 10.6% and theme from 4.9 to 6.7%, dropping the SFT adapter raises them to 12.4 and 7.3%, and dropping the retriever raises intent to 12.9% while cutting theme to 2.6%, the only clear cost any removal carries. Removing the critic changes the stopping decision and the per-step feedback at once, so the table cannot separate the direction head's information from the controller's enforcement of it. The two probes isolate both.

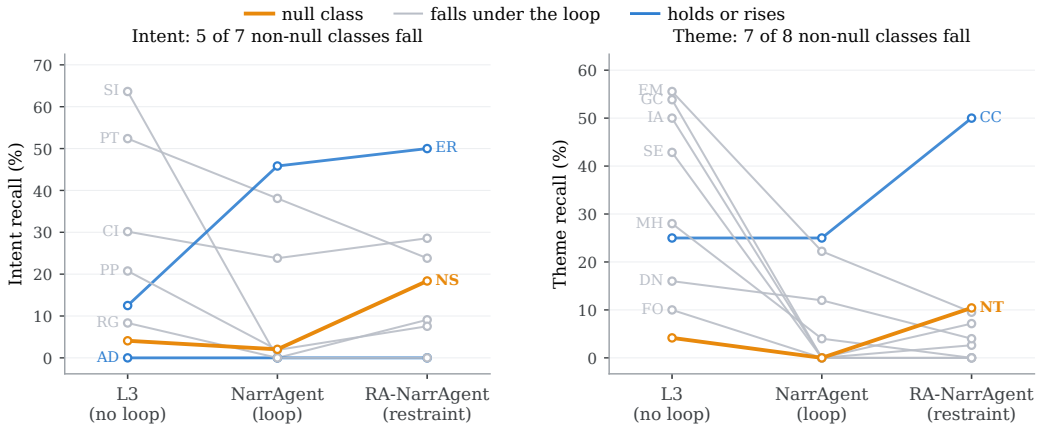


Figure 3: Per-class recall as the loop is switched on at a fixed prompt and then given the restraint mechanisms, one line per class. Recall rather than F1, because the claim is that whole classes stop being found. Falling lines are grey, the nulls orange, classes that hold or rise blue. The loop costs recall on most classes including the nulls, and RA-NarrAgent buys the nulls back by giving up PT and SI (Appendix D).

The two controller-replacement rows sharpen this. Removing the adapter while keeping the loop gives 12.2 and 4.0%, close to the leave-one-out arm removing the same adapter, while removing the controller's discretion instead, by calling all five tools in a fixed order, gives the best row in the table on every column, 15.0 and 13.6% macro-F1 at 16.7 and 19.2% accuracy. Both run on base weights at the same budget, so what separates them is whether the model chooses its own actions. SFT on trajectories carrying no variation in tool use or stopping behaviour (section 4.5) taught the controller to run the loop rather than answer the question.

5.4 Two Probes on the Critic and the Controller

Same input, different candidates. A critic seeing only the dialogue could learn "this input looks like NS, emit SIMPLIFY" without locating an unsupported claim, so the probe fixes the input and varies only the candidate, on 109 instances with four candidates each crossing correctness with depth, where "detailed" adds unlicensed claims. A candidate-blind critic would give all four the same direction. It does not, emitting SIMPLIFY on 12.8% of detailed against 4.6% of concise candidates at matched correctness ($+8.3$ points, 95% CI $[+5.8, +10.3]$, $p < 0.001$) and on none of the 109 when the candidate is withheld. Correctness nevertheless dominates depth, since wrong candidates draw SIMPLIFY at 50.5% against 4.6% ($+45.9$, $p < 0.001$) while within wrong candidates depth adds nothing detectable ($+0.9$, $[-5.2, +6.7]$). The direction head is substantially a correctness detector with a secondary sensitivity to over-assertion (Appendix G.1).

Advice against enforcement. The IDC enforces the critic's signal rather than offering it, so separating enforcement from information requires identical feedback under an *advisory* regime, where the signal enters the history and the agent may continue, against the *enforced* regime. Enforcement gives $+8.3$ points of intent accuracy (23.3% against 15.0%, $n = 60$ per arm) and lowers the false-positive rate on the corpus null instances the arms reach from 27.8% to 20.5%, but the difference is **not significant** ($[-3.2, +19.0]$, $p = 0.208$). The probe is weaker than designed, since the critic emitted zero SIMPLIFY signals in either arm, so the advisory agent never had a signal to override (Appendix G.2).

5.5 Cross-Model Replication

The pattern could be a quirk of one backbone, so L0, L3, NarrAgent and RA-NarrAgent are repeated with a Llama-3.1-8B-Instruct controller, holding instances, prompts, tools, trained critic and trained retriever fixed. The controller LoRA records the Qwen backbone as its base and cannot be loaded

on Llama, so these configurations run an *untrained* controller whose levels are not comparable to the Qwen row, and two backbones of similar scale do not rule out a family-specific effect, leaving only the within-row ordering interpretable. L3 still leads both axes, at 14.8% intent and 30.2% theme (Table 10). NarrAgent's columns are low largely because the untrained controller does not terminate, exhausting its budget on 60.8% of instances against 0.0% for the SFT'd Qwen controller, so that row corroborates the direction through a coarser cause. Against the Qwen pattern, RA-NarrAgent is the strongest Llama agent configuration on both axes, 12.3% intent and 12.7% theme against NarrAgent's 3.6 and 3.7%, at a non-termination rate of 10.0% rather than 60.8%, which reads as the IDC supplying the termination behaviour Qwen's controller acquired through SFT. Testing that needs an untrained Qwen controller, which was not run.

5.6 Sensitivity and Cost

The restraint mechanisms do not make the loop cheaper. RA-NarrAgent consumes 6,490 tokens per instance against 5,105 for NarrAgent and 1,157 for the prompt-only configuration that outperforms it, so the negative result holds on cost as well as accuracy (Appendix J). Sensitivity to θ, λ and T is not reported, since each sweep needs a full agent run per value, so all three are fixed a priori and the ablations test presence rather than tuning (Appendix C.4).

6 Per-Class Analysis

The per-class scores locate the loop's cost in the null classes (Figure 3). NarrAgent reaches 38.1% recall on PT, the most interpretively loaded intent class, and 45.8% on ER, against 2.0% on NS and 0.0% on NT, where L3 at the same prompt reaches 4.1 and 4.2%. The loop removes most of the little null recall the prompt-only configuration had and buys no interpretable-class recall in exchange, since PT falls from 52.4 to 38.1% and on theme every class except CC and DN drops. Only ER rises, so the loop concentrates its predictions on a few classes and loses the rest, which the macro-F1 gap between 18.2 and 8.8% records.

The concentration is onto one category rather than onto the frequent one. NarrAgent assigns CC to 123 of 240 instances and RA-NarrAgent to 148, against 4 instances that carry it and 5 assignments by L3, while neither over-predicts the modal class, EM at 63 instances, so this is not the accuracy-seeking collapse the imbalance would reward. Both collapse onto *Class Conflict*, attachable to almost any confrontational exchange, the category attractiveness bias of section 3.2. RA-NarrAgent moves the nulls back up short of prompt-only, to 18.4 and 10.4% on NS and NT, by giving up PT, which falls from 38.1 to 23.8% (Appendix D). No cross-system grounding comparison is reported, since that metric is a lexical-overlap check a system can raise by quoting more (Appendix I).

7 Discussion and Limitations

The phenomenon may extend to any pragmatic inference task handled by multi-component agents, adding to evidence that inference-time compute scaling is not monotonically beneficial (Gema et al., 2025; Chen et al., 2024), here through compound over-interpretation rather than the stagnation reported for mathematical reasoning. Four limitations bound the claims. Reference labels are machine-generated and reproduce at $\kappa = 0.474$ / 0.444, so only orderings are interpretable, each configuration ran once, local-gold scoring penalizes reinterpretation remote evidence may license, and the corpus is English-language Western screenplays in 12 test clusters with near-collinear nulls (Appendices B and A).

8 Conclusion

This paper documented configuration-dependent degradation in agentic subtext detection and asked whether explicit restraint undoes it. It does not. RA-NarrAgent recovers part of the intent gap short of significance and falls significantly below prompt-only on theme. The critic separates wrong from right candidates far better than inflated from restrained ones, and the deployed loop never emitted a restraint signal, so these numbers measure a missing mechanism as much as a failing one. Restraint must act on the gap between wrong and unlicensed, which this critic does not draw.

Statement on the Use of Large Language Models

A large language model was used to aid and polish the writing of this paper, covering grammar, wording and concision on text the authors had already drafted. The authors take full responsibility for the entire content of this paper.

References

Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. The art of scaling test-time compute for large language models. *arXiv preprint arXiv:2512.02008*, 2025.

Kabir Ahuja, Yuxuan Li, and Andrew Kyle Lampinen. Beneath the surface: Investigating LLMs' capabilities for communicating with subtext. *arXiv preprint arXiv:2604.05273*, 2026.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the International Conference on Learning Representations*, 2024.

Jiyun Bae, Hyunjong Ok, Sangwoo Mo, and Jaeho Lee. Understanding the effects of distractors on reasoning vision-language models. *arXiv preprint arXiv:2511.21397*, 2025. Formerly titled "Do Reasoning Vision-Language Models Inversely Scale in Test-Time Compute? A Distractor-centric Empirical Analysis"; to appear at EMNLP 2026.

Vidhisha Balachandran, Jingya Chen, Lingjiao Chen, Shivam Garg, Neel Joshi, Yash Lara, John Langford, Besmira Nushi, Vibhav Vineet, Yue Wu, and Safoora Yousefi. Inference-time scaling for complex tasks: Where we stand and what lies ahead. *arXiv preprint arXiv:2504.00294*, 2025.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1734–1752, 2021. doi: 10.18653/v1/2021.findings-emnlp.150.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. Story comprehension for predicting what happens next. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1603–1614, 2017. doi: 10.18653/v1/D17-1168.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8602–8615, 2022. doi: 10.18653/v1/2022.acl-long.589.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*, 2024.

DeepSeek-AI, Daya Guo, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Aryo Pradipta Gema, Alexander Hägele, Runjin Chen, Andy Arditi, Jacob Goldman-Wetzler, Kit Fraser-Taliente, Henry Sleight, Linda Petrini, Julian Michael, Beatrice Alex, Pasquale Minervini, Yanda Chen, Joe Benton, and Ethan Perez. Inverse scaling in test-time compute. *Transactions on Machine Learning Research*, 2025. Featured Certification. `https://openreview.net/forum?id=NXgyHW1c7M`.

Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1066–1076. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1113.

H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and Semantics, Volume 3: Speech Acts*, pp. 41–58. Academic Press, 1975.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the International Conference on Machine Learning*, 2020.

Shreya Havaldar, Hamidreza Alvari, John Palowitch, Mohammad Javad Hosseini, Senaka Buthpitiya, and Alex Fabrikant. Entailed between the lines: Incorporating implication into NLI. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32274–32290. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1552.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *Proceedings of the International Conference on Learning Representations*, 2024.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1534–1544, 2016. doi: 10.18653/v1/N16-1180.

Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. Are natural language inference models IMPPRESsive? learning IMPlicature and PRESupposition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8690–8705, 2020. doi: 10.18653/v1/2020.acl-main.768.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5), 2017. doi: 10.1145/3124420.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. doi: 10.1162/tacl_a_00713.

Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Le Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. Fantom: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 14397–14413, 2023. doi: 10.18653/v1/2023.emnlp-main.890.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, pp. 6536–6558, 2022. doi: 10.18653/v1/2022.findings-emnlp.488.

David Y. Liu, Aditya Joshi, and Paul Dawson. Narrative theory-driven LLM methods for automatic story generation and understanding: A survey. *arXiv preprint arXiv:2602.15851*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-2019.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. Inverse scaling: When bigger isn't better. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=DwgRm72GQF.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 839–849, 2016. doi: 10.18653/v1/N16-1098.

OpenAI. Learning to reason with LLMs. OpenAI research blog, 2024. URL https://openai.com/index/learning-to-reason-with-llms/. Published 12 September 2024.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie plot analysis via turning point identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1707–1717, 2019. doi: 10.18653/v1/D19-1180.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. The goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by LLMs. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-0913. Earlier version titled "Large Language Models Are Not Zero-Shot Communicators".

Yisi Sang, Xiangyang Mou, Mo Yu, Shunyu Yao, Jing Li, and Jeffrey Stanton. TVShowGuess: Character comprehension in stories as speaker guessing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4267–4287, 2022. doi: 10.18653/v1/2022.naacl-main.317.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-0377.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint arXiv:2402.08115*, 2024.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in LLMs. *arXiv preprint arXiv:2505.00127*, 2025.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of the International Workshop on Semantic Evaluation*, pp. 39–50, 2018. doi: 10.18653/v1/S18-1005.

Yue Wan, Xiaowei Jia, and Xiang Lorraine Li. Unveiling confirmation bias in chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3788–3804. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.195.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), 2024. doi: 10.1007/s11704-024-40231-1.

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

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

An Yang et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations*, 2023.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the International Conference on Machine Learning*, 2024.

A Dataset Details

LitSubText comprises 2,606 dialogue instances drawn from 132 film screenplays. Screenplays were selected to span a range of genres (drama, thriller, comedy, romance, science fiction, and historical) and time periods (1950s through 2020s), with a preference for films widely regarded as having rich subtextual dialogue. All screenplays were sourced from publicly available screenplay databases.

Each instance consists of 3–8 dialogue turns copied verbatim from the screenplay, one of which is designated the target utterance, together with a scene description and a local context summary. The exchange was selected by the annotating model from a $\sim$3,000-character window of raw screenplay text, not segmented by hand; the window boundaries are character offsets with 300-character overlap, broken at blank lines, so they do not respect scene boundaries. Dialogue turns are copied rather than paraphrased, but the scene description and local context are written by the annotator and are not verbatim text.

Screenplay selection. Screenplays were obtained from IMSDb, a public screenplay archive, using a fixed list of candidate titles. Selection favored films with critical recognition for dialogue and aimed for spread across genre and decade, but it was not a rated or blinded procedure: there was no independent subtextual-richness rating, no exclusion threshold, and no second-round filtering. Titles that failed to download or parse were dropped, leaving 132 screenplays. The resulting corpus is therefore a convenience sample of what the archive carries in parseable form, weighted toward well-known English-language films, and its genre labels are assigned by a keyword rule over titles rather than by human coding.

The null quota. The collection pipeline is given as six stages below. One of its mechanisms shapes the corpus enough to isolate here. The default instruction asks for an exchange carrying genuine subtext, at a target of roughly 75–80% such instances. Once a screenplay is close to its 20-instance limit with fewer than four null instances collected, the instruction switches, and the model is asked for an exchange in which the characters speak *directly and literally*, setting presence to false with intent NS and theme NT. This produces the observed null rate of about 20%, and explains two properties of the data. The null rate is near-constant in every partition of the corpus by screenplay, holding between 17.9 and 21.2% across genres and 19.9 to 20.3% across decades, because the quota applies per screenplay. And NS and NT are near-collinear, with 518 of 521 NS instances also NT and no instance NT alone, because the switched instruction assigns both null labels in one breath rather than judging the two axes separately. Null instances are therefore selected under a different instruction than non-null ones, so the two groups are not exchangeable: a null instance is one the annotator found when asked to find something literal.

Label taxonomy. Both taxonomies were specified by the authors before annotation, drawing on the pragmatics and screenwriting literature cited in section 3.1, and were held fixed throughout collection. There was no pilot annotation round, no merging of low-agreement categories, and no empirical refinement of the class inventory: the 8 intent and 9 theme classes are the ones written into the annotation prompt from the start. The null classes were included by design, on the reasoning that a classifier forced to choose among substantive categories cannot express insufficient evidence, the same argument the paper makes about category attractiveness in agents, rather than in response to an observed annotation difficulty.

This matters for how the taxonomy's fit should be read. Because the classes were never revised against annotation data, the agreement figures in Appendix B are the first evidence about whether

the distinctions are reliably applicable, and they show the fit is uneven: SI is recognized consistently across annotators, while NS and RG are not, and on the theme axis EM absorbs a large share of instances and is among the least reliably distinguished. A taxonomy developed through iterative agreement-driven refinement would likely have merged or re-specified those classes. We report the inventory as designed, with the per-class reliability alongside it, rather than presenting it as validated.

Table 3 provides definitions and examples for each intent class.

Table 3: Intent class definitions with one corpus instance each, quoted verbatim from the target utterance with its screenplay.

Code	Definition	Example
PT	Probing / Testing: speaker seeks information or gauges a reaction without asking directly	"I go to a lot of therapy, Ronnie. What are you trying to say?" *Silver Linings Playbook*
ER	Evasive Refusal: speaker declines without stating a refusal	"There's nothing to tell. It's been stricken from the record." *The Verdict*
CI	Concealment / Insincerity: speaker hides knowledge, intent or true feeling	"It's getting published. That's what we're up here celebrating." *Sideways*
SI	Sarcasm / Irony: intended meaning inverts or undercuts the literal one	"It teaches me a valuable lesson… Apparently crime pays." *Traffic*
PP	Provocation / Pressure: speaker pushes, needles or coerces	"Clarice… you're hardly in a position to accuse me of lying." *Silence of the Lambs*
AD	Appeasement / De-escalation: speaker concedes or softens to defuse conflict	"I'm saying if you want. If that's what you want, I'll do it." *25th Hour*
RG	Resentment / Grievance: speaker voices accumulated complaint indirectly	"So you run with the American's now, huh? Now times are bad?" *Inglourious Basterds*
NS	No Strong Subtext: insufficient evidence for non-literal reading	"It's 90 degrees, there ain't been a cloud in the sky in weeks." *Bull Durham*

Dataset statistics. The dataset is split at the screenplay level so that no screenplay appears in more than one split: 108 screenplays for training (2,126 instances), 12 for validation (240 instances), and 12 for testing (240 instances). The entire test split is the evaluation set; nothing is held back. The intent class distribution is: CI (23.6%), PP (22.6%), NS (20.0%), ER (12.5%), PT (7.0%), SI (6.7%), RG (4.7%), AD (2.8%). The theme class distribution is: EM (25.7%), IA (20.9%), NT (19.9%), DN (8.6%), MH (8.1%), SE (5.7%), FO (4.9%), GC (4.1%), CC (2.1%). The distribution is markedly non-uniform on both axes, which is why accuracy is reported against a majority-class floor rather than a uniform-chance floor (section 5). The null intent label NS covers 20.0% of the corpus, and the null rate is stable across splits, at 19.9% on train, 20.0% on validation and 20.4% on test. Per-class test shares track the corpus to within 5 points on every class, the largest deviations falling on the rarest ones, where a single screenplay's habits move the count: IA is 20.9% of the corpus and 15.8% of the test split, SI 6.7% against 4.6%.

Genre distribution. Drama is the largest category at 49 screenplays and 964 instances, followed by Comedy and Crime at 19 screenplays each, then Historical (16), Romance (13), Sci-Fi (8), and Thriller and War (4 each). That ordering follows from how the corpus was assembled rather than from a measured property of any genre. Genre labels come from a keyword rule over film titles, not from an external database or human coding, so they should be read as a coarse partition for auditing coverage; a film matching several rules takes the first match. The per-genre NS and NT rates are flat, spanning 17.9–21.2% across genres, which is expected: the null rate is set by a per-screenplay quota applied uniformly during construction (§B) rather than emerging from the dialogue, so it carries no information about genre. We report the breakdown so composition is auditable; the corpus was not sampled to support comparisons between genres and we draw none.

Decade distribution. The corpus spans nine decades, concentrated in the 1990s (31 screenplays, 618 instances) and 2000s (37, 724), which reflects the availability of digitized screenplays; the 1940s through 1960s contribute 7 screenplays and 140 instances between them. Mean turns per instance sits between 5.1 and 5.8 in every decade and mean speakers at 2.1 to 2.2, so dialogue shape is flat across the range and the NS rate holds at 19.9 to 20.3% throughout, again by construction. Release

years are resolved from a title-to-year lookup that covers all 132 screenplays. We report the decade breakdown for coverage rather than as an object of analysis: the corpus is not a sample designed to support claims about how dialogue conventions changed over time, and the per-decade counts, 20 instances in the 1960s and 40 in the 1940s, are far too small and too confounded with genre to license them.

NS prevalence is flat across decades, between 19.8% and 20.3% in every bucket. This is a property of the construction procedure rather than a finding about film history: the null quota was applied per screenplay at a fixed rate, so the null rate is approximately constant within any partition of the corpus by screenplay. The instance counts are likewise a construction artefact, at 20 per screenplay for all but three films. Neither table supports an inference about how dialogue conventions have changed, and a corpus built to test that question would need the null rate to be free to vary.

Dialogue length statistics. Instances average 6.8 turns (median 6, range 2 to 20) and 98.4 whitespace-delimited tokens (median 82, range 12 to 412), with 2.4 distinct speakers and 142 words of scene context. Both length measures are right-skewed, the mean sitting above the median in each case. We truncate instances at 20 turns for computational tractability, affecting fewer than 3% of instances.

Data collection pipeline. The data collection process consisted of six stages, all automated after the title list was fixed.

Screenplay sourcing. The corpus is built from a fixed hand-written list of 137 IMSDb script paths (136 distinct titles; one duplicate entry), assembled by the authors with a preference for dialogue-driven films with recognition for screenplay quality. There was no expert review step and no formal inclusion criterion beyond that preference, so the title list should be read as a convenience sample rather than a stratified one. Four titles yielded no instances, because the fetch or the parse failed, leaving 132 screenplays and 2,606 instances. Genre labels are not curated either: they come from a keyword rule over the title string, taking the first match when several apply.

Preprocessing. No structural parse is used on the labeling path. The screenplay is fetched as plain text (the `<pre>` block of the IMSDb page with HTML tags stripped, falling back to the whole body) and passed to the chunker as raw text; a scene-level parser exists in the codebase but the corpus was not built through it. Speaker attribution, scene boundaries, and scene descriptions are therefore produced by the annotating model reading raw text, not by a parser, and we report no parser accuracy figures because no parser stands between the source text and the annotator.

Chunking. Each parsed screenplay is split into overlapping windows of roughly 3,000 characters with a 300-character overlap, breaking at blank lines so that a window boundary does not fall inside a speech. Windows are visited in a shuffled order, which prevents the instances from any one screenplay from all coming from its opening scenes.

Selection and labeling in one pass. A single LLM call per window both selects the exchange and labels it: given the raw window, the model extracts one dialogue exchange verbatim, designates a target utterance, and emits the intent label, theme label, presence judgment, an explanation, and evidence spans. Windows with no usable dialogue return an explicit error and are skipped. Selection and labeling are therefore not independent, since the same pass that assigns the label also decides which exchange becomes an instance, a limitation we return to below.

Null-class quota. Because a model asked for "the best exchange" in a window systematically prefers interpretively rich material, null instances would otherwise be nearly absent. We therefore reserve a per-screenplay quota of windows for which the prompt instead asks for an exchange where characters speak directly and literally. This is what produces the 19.9% null rate; without it the null class would be too rare to support the FPR and null-recall measurements that the paper's central claim rests on.

Retrieval pool. For each instance, the three other windows of the same screenplay with the highest keyword overlap with the target utterance are attached as retrieved passages, falling back to adjacent windows when overlap is zero. The retrieval corpus is thus within-screenplay by construction.

Label provenance and its consequences. The labels are machine-generated. Each instance carries the output of one `claude-sonnet-4-6` call under the taxonomy definitions of section 3.1;

there is no multi-annotator vote, no adjudication step, and no human-authored gold label. Consequently there is no candidate pool that was filtered down by agreement: every successfully parsed annotation is in the released set, and the 2,606 instances are the 2,606 that returned well-formed output.

This has three consequences we ask readers to hold in view. First, the "gold" labels are a single model's judgments and inherit its biases, including whatever preference it has among the interpretive categories; the class distribution reported above should be read as a property of the annotator, not of screenplay dialogue in general. Second, because the same pass selected and labeled each exchange, the dataset is not a random sample of film dialogue but a sample of exchanges one model found label-worthy. Third, agreement figures cannot be computed within the annotation process itself, since there is only one annotator. We therefore report reliability externally, by comparing the released labels against independent annotators on a 200-instance subset (Appendix B), and we interpret accuracy against a majority-class floor throughout.

We regard this as the principal limitation of LitSubText as a benchmark, and it bounds what the paper's accuracy numbers mean: a system is being scored on its agreement with one model's reading, not against a human consensus. The overthinking phenomenon itself is more robust to this concern than the absolute accuracies are, because it is a comparison *between* systems scored against the same labels, and a label set that is noisy but fixed cannot by itself manufacture a non-monotonic accuracy curve across configurations, but the absolute values should be read as agreement rates rather than correctness rates.

Comparison with related datasets. Against existing narrative NLP corpora, LitSubText is distinguished by its focus on pragmatic interpretation rather than surface comprehension, its multi-axis scheme, and its null class. Of NarrativeQA, BookSum, SummScreen, StoryCloze, TVShowGuess and FANToM, only FANToM targets a pragmatic task, and none carries a null class.

The most important distinction is the presence of the null class and the pragmatic nature of the task. Most existing datasets assume that every instance has a correct non-trivial answer; LitSubText explicitly includes instances where the correct answer is "no subtext is present," making the task more realistic and enabling evaluation of restraint-related failure modes.

B Label Provenance and Reliability

This appendix states how the LitSubText labels were produced and what evidence we have about their reliability. We report it in detail because the labels are machine-generated, which bounds the interpretation of every accuracy number in the paper.

The annotation procedure. Each instance was labeled by a single call to `claude-sonnet-4-6`. The model received one ∼3,000-character window of raw screenplay text together with the taxonomy definitions of section 3.1, and returned, in one structured response: an exchange extracted verbatim from the window, a designated target utterance, the intent label, the theme label, a binary presence judgment, a free-text explanation, and evidence spans quoted from the dialogue. There was no second pass, no vote across annotators, and no adjudication. The NS and NT definitions given to the annotator are the evidence-threshold ones of section 3.1: a null label asserts that the visible window provides insufficient evidence for a specific reading, not that no subtext exists in the film.

Because an annotator asked to find the most interesting exchange in a window will systematically prefer interpretively rich material, a per-screenplay quota of windows instead requested an exchange in which characters speak directly and literally. This quota is the mechanism that produces the 19.9% corpus null rate. It also means null instances were selected under a different instruction than non-null ones, so the two groups are not exchangeable: null instances are, by construction, ones the annotator judged literal when asked to find something literal.

With one annotator, no agreement statistic can be computed inside the annotation process. Fleiss' κ, pairwise κ across annotator pairs, vote distributions, modal share, and adjudication rates are all undefined here, and we report none of them. Nor can we appeal to annotator training or domain expertise as a quality argument, and we do not: the taxonomy was applied by a general-purpose

model under a fixed prompt, and its consistency is whatever that setup yields. The reliability question we *can* address is external, namely whether an independent annotator, given the same taxonomy and the same visible context, reproduces these labels, and that is what the rest of this appendix measures.

B.1 External Reliability Through Cross-Annotator Agreement

Design. We drew 200 instances from the corpus and had them re-annotated independently by two further annotators from different model families, GPT-5.5 and Llama-3.3-70B. Each received the same dialogue, the same visible context, and the same taxonomy definitions as the original pass, with no access to the released label, the explanation, or the evidence spans. This gives three independent label vectors per instance on all three axes and 200 usable triples with no parse failures and no off-taxonomy labels. The subset preserves the corpus mix closely: released intent labels are PP 23.5%, NS 22.5%, CI 21.5%, ER 12.0%, SI 8.5%, PT 6.5%, RG 4.0%, AD 1.5%.

The human condition. Six annotators labeled the 200 instances under a blind protocol, three assigned to items 1–100 and three to items 101–200, giving three independent human judgments per item and 600 in total. Released labels, machine-written explanations and evidence spans were withheld, and interpretive sentences were stripped from the scene descriptions so the context field could not state the answer, which affected 129 sentences across 94 of the 200 items, some naming the gold label outright. Items were ordered so that every prefix preserves the corpus class proportions rather than being class-balanced, since κ depends on the marginal distribution, making a truncated run a stratified sample. The annotators are native Chinese speakers, so each English line is followed by a Chinese rendering with the instructions directing all judgments to the English original; the renderings were produced under a surface-literal prompt and screened for vocabulary that resolves the subtext rather than carrying it over, since a translation that names the speaker's move supplies the label.

Human-human agreement is computed within each half, over the six annotator pairs that share one. One annotator also recorded 20 judgments in the other half, which are dropped so that every annotator contributes the same 100 items. Three single-cell responses fell outside the taxonomy and are dropped from the affected axis, and one annotator transposed the intent and theme columns on rows marked null, recognizable because the pair reads (NT, NS) rather than (NS, NT); those rows are corrected before scoring. No other repair is applied.

Scope of the human agreement figures. Humans reproduce the released intent label at $\kappa = 0.177$ and agree among themselves at $\kappa = 0.181$, so the shortfall against the model annotators is a property of the 8-way distinction rather than of the released labels in particular: a label set that human readers merely disagreed with would show them agreeing with each other. Agreement rises as the decision coarsens, reaching $\kappa = 0.352$ for the null versus non-null contrast, and the null class is the one humans recover most often at 53.6% against 12.8% for CI. The presence axis behaves the same way, at $\kappa = 0.364$ against the released labels and 0.361 between annotators. This bounds the fine-grained numbers and leaves the paper's claims intact, since those rest on the null contrast and on comparisons between systems scored against one fixed reference. Absolute per-class accuracies on the substantive intent classes should be read as agreement with the reference annotator.

Agreement between annotators. Table 4 reports Cohen's κ and raw agreement for all three annotator pairs on all three axes.

Intent agreement with GPT-5.5 is moderate at $\kappa = 0.474$, and theme agreement likewise at 0.444; agreement with Llama-3.3-70B is fair on intent and poor on theme. Two implications matter more than those conventional bands. First, the ceiling is low. If an independent annotator reproduces the reference at $\kappa \approx 0.47$ on intent, a system scored against that reference should not be expected to exceed the agreement two annotators reach with each other, and the absolute accuracies reported in section 5 are much closer to that ceiling than to 100%. Second, the axis with the weakest agreement is presence, and presence is the axis the restraint claim is scored on. We return to this below.

Source of the low Llama-3.3 column. The Llama-3.3 figures should not be read as evidence that the taxonomy is unusable, because that annotator's own label distribution is degenerate: it assigns CI to 106 of 200 instances (53.0%), against 21.5% in the released labels and 22.0% for GPT-5.5,

Table 4: Cross-annotator agreement on the 200-instance reliability subset. "Released" is the label shipped with LitSubText, κ is Cohen's κ and raw is the proportion of exact matches. The three model pairs have $n = 200$ with no dropped instances. The human row pools 598 intent and 599 theme judgments from six annotators of 100 items each, and the human-human row averages κ over the six annotator pairs sharing a half.

	Intent		Theme		Presence	
Pair	κ	raw	κ	raw	κ	raw
Released vs. GPT-5.5	0.474	0.550	0.444	0.545	0.334	0.830
Released vs. Llama-3.3-70B	0.361	0.465	0.231	0.330	0.131	0.795
GPT-5.5 vs. Llama-3.3-70B	0.426	0.525	0.252	0.360	0.519	0.965
Released vs. human (pooled)	0.177	0.289	0.155	0.294	0.364	0.768
Human vs. human (mean)	0.181	0.302	0.145	0.306	0.361	0.757

and assigns ER to 4 instances against 24 released. Its NS rate is 3.5%. A single class absorbing half the sample depresses κ against any reference, and its largest disagreements are exactly what that collapse predicts: 21 of 47 released PP and 21 of 45 released NS instances become CI. GPT-5.5's distribution is close to the released one on intent, and its disagreements are dispersed rather than concentrated in one class. We therefore treat the released-vs.-GPT-5.5 row as the informative estimate of taxonomy reliability and the Llama-3.3 row as a demonstration that annotator quality varies sharply across models, which is itself relevant, since LLM-as-annotator is the provenance of these labels.

Notably, the two independent annotators agree with each other on presence at $\kappa = 0.519$, higher than either agrees with the released labels. Both mark far fewer nulls than the reference, so they agree substantially on a low-null reading of the corpus while both diverging from the reference; agreement between annotators and agreement with the reference come apart here.

Per-class agreement. Table 5 reports one-vs-rest κ per intent class, computed by recoding each instance as "this class" versus "any other" for both annotators. One-vs-rest keeps every class on the same 200 instances with the same annotators; the alternative, restricting to instances whose reference label is the class in question, selects the sample by the very label being evaluated, and cannot separate a class that is hard to recognize from one that is rare.

Table 5: Per-class one-vs-rest Cohen's κ between the released labels and each independent annotator, on all 200 instances of the reliability subset. n is the number of instances carrying the class in the released labels. The disagreement columns give the single most frequent alternative label assigned to instances the reference marks as this class.

		One-vs-rest κ		Modal disagreement	
Class	n	vs. GPT-5.5	vs. Llama-3.3	vs. GPT-5.5	vs. Llama-3.3
PP	47	0.513	0.439	SI (4)	CI (21)
NS	45	0.322	0.222	AD (12)	CI (21)
CI	43	0.545	0.333	AD (6)	AD (2)
ER	24	0.541	0.260	CI (5)	CI (16)
SI	17	0.688	0.616	PT (1)	RG (4)
PT	13	0.576	0.407	PP (2)	CI (4)
RG	8	0.201	0.430	ER (2)	PT (1)
AD	3	$n = 3$*; not reported*			

The pattern is that agreement tracks how observable the class is in the dialogue itself. SI is the most reliably recognized class ($\kappa = 0.688$ / 0.616), and it is the one whose cue, an utterance whose surface content is at odds with an obvious shared fact, is locally visible. CI and ER, both moderate against GPT-5.5, likewise rest on observable deflection or elicitation. NS is among the weakest (0.322), and RG is low against GPT-5.5 at 0.201 on 8 instances, where the estimate is unstable. We do not report a coefficient for AD, which has 3 instances in the subset: a κ on 3 instances is not

an estimate of anything, and reporting one would invite the over-reading the paper argues against elsewhere.

On the theme axis the same ordering holds and the spread is wider. Against GPT-5.5, agreement runs from GC 0.745 and DN 0.536 and MH 0.521 down to EM 0.402 and SE 0.404, with CC at -0.019 on 3 instances. EM is both the largest class (55 of 200 released, and 82 of 200 for GPT-5.5) and among the least reliably distinguished, which suggests it functions partly as a default when no sharper theme applies. Against Llama-3.3 the theme coefficients largely collapse (MH 0.063, SE 0.044, GC 0.000), consistent with that annotator's concentration on IA and DN.

The null boundary. The null classes deserve separate treatment, because a null label asserts the absence of attributable content and is not symmetric with the substantive classes. Of the 200 subset instances, 45 carry NS. Both independent annotators assign NS to 6 of them, exactly one does for a further 6, and neither does for the remaining 33. The reverse direction is nearly empty: GPT-5.5 marks NS on 1 instance the reference treats as non-null, Llama-3.3 on none. The disagreement is thus almost entirely one-directional, since the reference asserts a null where independent annotators find a specific intent, at rates of 22.5% versus 6.0% and 3.5%.

This is the appendix's most consequential finding for the paper, and it cuts in a specific direction. It is not the case that the reference is unusually willing to read subtext in; it is unusually willing to decline to. Two readings are available and we cannot fully separate them. The quota instruction may have made the reference annotator systematically stricter about the evidence threshold than an annotator given only the definitions, in which case the reference null rate is the better-calibrated one and the independent annotators are over-reading. Alternatively the reference may mark literal-looking exchanges as null that do carry recoverable intent, in which case the null labels are partly an artefact of how those windows were sampled. The one-directionality is consistent with either, since both predict nulls concentrated in the reference.

What follows regardless is a caution about FPR. The false-positive rate on null instances is defined against the axis where annotators agree least ($\kappa = 0.334$ / 0.131), and against a reference whose null rate is three to six times either independent annotator's. A system's measured FPR therefore carries real uncertainty about the denominator, and FPR differences between systems are more trustworthy than any single FPR value, the same argument the paper makes for accuracy, but with a wider margin. Where the appendix does sharpen the picture is in the kind of disagreement. Cases where annotators agree the dialogue is unremarkable but differ on whether a faint interpretive residue clears the threshold are threshold calibration. Cases where annotators agree something is marked but differ on whether it is attributable to the speaker's intent or is a property of the scene's framing are a taxonomy boundary between NS and the thematic axis. A system failing the first is miscalibrated; one failing the second is attributing scene-level content to speaker intent, which is the specific failure section 3.2 labels H2.

Consequences for the benchmark. We state these plainly rather than deferring them. LitSubText measures agreement with one model's application of an interpretive taxonomy, at a reliability level where a well-behaved independent annotator reproduces the intent label at $\kappa = 0.474$. This supports the paper's central comparison, which is *between* systems scored against a fixed reference: a noisy but fixed label set cannot by itself produce a non-monotonic accuracy curve across configurations, so the overthinking pattern does not reduce to label noise. It does not support treating any absolute accuracy as a correctness rate, drawing per-class conclusions on AD or CC, or claiming the taxonomy has been validated against human judgment. The last would require a multi-annotator human study with adjudication; the single-annotator pilot described above is a first step toward it and not a substitute for it.

C Implementation Details

C.1 The RA-NarrAgent Procedure

Algorithm 1 gives the complete loop described in section 4.4.

Algorithm 1 RA-NarrAgent procedure

Require: Dialogue d, scene s, step budget T, CGR threshold θ, post-MAINTAIN allowance $T_m \geq 1$, reward floor τ_r

$interp \leftarrow \varnothing$; $dropped \leftarrow \varnothing$; $history \leftarrow [(d, s)]$; $used \leftarrow 0$; $cap \leftarrow T$
Inject NFP into system prompt {null-first default}
while $used < cap$ **do**
  $\mathcal{A} \leftarrow \{\text{RETRIEVE}, \text{ANALYZE}\}$; **if** $interp \neq \varnothing$ **then** $\mathcal{A} \leftarrow \mathcal{A} \cup \{\text{VERIFY}, \text{CONCLUDE}\}$
  $action \leftarrow \text{LLM}(history,\ \mathcal{A})$; $used \leftarrow used + 1$ {every LLM call is charged}
  if $action = \text{RETRIEVE}(q)$ **then**
    $P \leftarrow \text{CGR}(q, \theta)$ {section 4.2; returns $\varnothing$ if below θ}
    Append P (or "no relevant passages") to $history$
  else if $action = \text{ANALYZE}$ **then**
    $interp \leftarrow \text{LLM}(history,$ "structured decomposition"$)$; $used \leftarrow used + 1$
    Append $interp$ to $history$
  else if $action = \text{VERIFY}$ **then**
    $r,\ (axis, dir) \leftarrow \text{BiCritic}(d, s, interp)$ {dir is a depth signal, not a correctness verdict}
    Append $(r, axis, dir)$ to $history$ {feedback available to LLM}
    if $dir = \text{SIMPLIFY}$ **then**
      $interp,\ dropped \leftarrow \text{Revise}(history, axis)$; $used \leftarrow used + 1$
      break {revision is the final interpretive act}
    else if $dir = \text{MAINTAIN}$ **then**
      if $r < \tau_r$ **then**
        $interp \leftarrow \text{Reselect}(history)$; $used \leftarrow used + 1$ {relabel from existing claims}
      end if
      $cap \leftarrow \min(cap,\ used + T_m)$ {T_m further steps, then stop}
    end if
  else if $action = \text{CONCLUDE}$ **then**
    break
  end if
end while
if $interp = \varnothing$ **then**
  $interp \leftarrow \text{LLM}(history,$ "structured decomposition"$)$; $used \leftarrow used + 1$
end if
return $\text{Emit}(interp,\ dropped)$ {formats the settled labels; may not reintroduce $dropped$}

C.2 CONFIGURATION DEFINITIONS

Table 6 specifies the full set of variables that differ across configurations L0–A. Each level changes one or more dimensions relative to the previous; the configurations are not strictly nested.

Table 6: Configuration definitions. Temp. is decoding temperature; Cands. is the number of candidates for selection. Each row's input is cumulative over the row above, matching the prompt templates in Appendix F: the target utterance is marked in every configuration including L0, so it is not a difference between levels.

Config	Input	Retrieval	Temp.	Cands.	Selection
L0	dialogue only	—	0	1	direct
L0+local	+ scene context	—	0	1	direct
L0+retr.	+ retrieved passages	top-3	0	1	direct
L3	+ narrative-context field	top-3	0	1	direct
L3+NFP	= L3, null-first prompt	top-3	0.3	1	direct
NarrAgent+NFP	ReAct loop, $T = 6$, null-first	top-3	0.7	1	agent decision
NarrAgent-Short	ReAct loop, $T = 3$, no NFP	top-3	0.7	1	agent decision
NarrAgent+Scalar	$T = 6$, NFP, no dir. head	gated	0.7	1	agent decision
RA-NarrAgent	$T = 6$, NFP, CGR, BiCritic, IDC	gated	0.7	1	IDC
Ablation-only arms (Table 2), base weights, $T = 6$, no adapter					
Fixed pipeline	five tools in fixed order, no choice	top-3	0.3	1	all observations
Zero-shot agent	ReAct loop, no SFT adapter	top-3	0.7	1	agent decision

LLM backbone and decoding. All configurations use the same backbone, stated in the experimental setup, so that no comparison confounds capability with scale. The four context-only settings decode greedily (temperature 0), where the output is a single sample and determinism is free. L3+NFP runs at 0.3 and the agent configurations at 0.7: a loop that samples its own next action deterministically can lock into a repeated action. None of these values was tuned, and all configurations cap generation at 300 new tokens.

Retrieval is *top-3 unconditional* in the context-only settings and in NarrAgent+NFP and NarrAgent-Short, and *gated* at $\theta = 0.55$ in NarrAgent+Scalar and RA-NarrAgent, since the gate is the mechanism section 4.2 introduces, so only those two configurations have it. Two consequences are worth stating rather than leaving to the table. First, NFP is present in every agent configuration except NarrAgent-Short, so the comparison against L3+NFP is the one that isolates the loop while holding the prompt fixed; there is no agent configuration without NFP, and we do not claim one. Second, NarrAgent+Scalar differs from RA-NarrAgent in two respects at once, dropping the direction head *and* the IDC that consumes it, so the control probe of section 5.4, which varies enforcement alone, is what separates them. The leave-one-out rows of section 5.3 do not, because they are relative to NarrAgent at a 4-step budget rather than to RA-NarrAgent. We report no comparison against external agent frameworks; section 5 explains why the needed comparison is internal.

Retriever and BiCritic. The dense retriever is initialized from `bge-m3` and fine-tuned with InfoNCE over relation-aware hard negatives, and the BiCritic is Qwen2.5-3B with LoRA adapters carrying two heads, a scalar reward and a three-way direction classifier. Every hyperparameter for both is listed in Table 9 with how it was chosen; the column matters more than the values, since only the retriever learning rate and the epoch counts were selected on validation loss and everything else is a default or a figure set by inspection. Two choices do need stating in prose because the table cannot express them. The retriever is trained on within-screenplay passages only, so it never learns to discriminate across films, which is the setting it is evaluated in. And the direction head is trained on the tier structure of section 4.3 rather than on observed agent behaviour, with Table 7 giving the tier-to-label map, so its three labels describe what a candidate looks like, not what the agent should do next, which is the gap §5.4 probes.

Table 7: Direction label assignment for BiCritic training, per axis. "Null axis" means the gold label on the axis being modified is NS (intent) or NT (theme).

Tier	Non-null axis	Null axis
A (gold)	MAINTAIN	MAINTAIN
B (correct, thin support)	DEEPEN	MAINTAIN
C-over (wrong via excess)	SIMPLIFY	SIMPLIFY
C-under (wrong via surface)	DEEPEN	DEEPEN
D (literal)	DEEPEN	MAINTAIN
E (over-interpretation)	SIMPLIFY	SIMPLIFY

For reward pairing, Tier E serves as a strong negative. On a null instance the margins are $m(A, E) = 0.75$ and $m(B, E) = 0.5$, so an interpretation that asserts concealed content where gold denies it is ranked below both the gold reading and a shallow-but-correct one. The same two margins apply on a non-null instance, where the remaining gaps are $m(A, B) = 0.25$, $m(A, C) = 0.5$, $m(A, D) = 1.0$ and $m(B, D) = 0.5$. On a null instance $m(A, B)$ and $m(A, D)$ drop to 0.15, since a literal paraphrase of a literal passage is close to correct and should not be pushed far down, which inverts the relative position of D and C without either pair being formed directly. Two $A \succ C$ pairs are drawn per instance with independently sampled wrong labels.

Tier E is generated for instances whose gold intent falls in the restrained set {NS, PP, AD}, which is where an inflated reading is unambiguously wrong. It is not generated for instances whose gold label is already an interpretively attractive class, because there the inflated label and the correct label can coincide and the pair would no longer isolate over-assertion. This is a deliberate restriction with a cost we state plainly: the critic never sees an over-interpretation example built on a CI, PT, SI, ER, or RG instance, so *on those classes* restraint supervision is absent, and a critic that learned "restrained classes attract SIMPLIFY" would fit this data. §5.4 probes exactly that confound by holding the instance fixed and varying only the candidate.

We also state a divergence between the intended design and the implementation. The design calls for Tier E to modify one axis at a time, leaving the other at its gold reading, so that a margin is attributable to a single claim. The implementation replaces both labels simultaneously, intent drawn from {PT, SI, CI} and theme from {EM, DN, CC}, and supervises both direction heads with SIMPLIFY. Because NS and NT co-occur on 518 of 521 instances in this corpus (section 3.1), the per-axis and merged constructions almost always coincide here, and the single-axis condition the design contemplates is nearly unrealizable: only 3 instances are NS without being NT and none are NT without being NS. The consequence is that the critic cannot learn to penalize an inflated intent claim while preserving a correct thematic reading, since no training pair separates them. A corpus with decoupled null axes would be needed to train and test that distinction, and none is currently available: in LitSubText 518 of the 521 NS instances are also NT.

Table 8 reports the resulting counts.

Table 8: Critic training pairs by tier gap, on the real screenplay-level splits (seed 42). Each Tier E instance contributes two pairs, (A, E) and (B, E), so the A–E column also gives the number of instances carrying a Tier E example: 959 of 2,126 training instances (45.1%). The A–C row is doubled because two distinct wrong-label distractors are drawn per instance. Only pairs from the train split are used for critic training; the val and test columns are reported for completeness.

Tier gap	Train	Val	Test	Rejected candidate is
A–B	2,126	240	240	correct but shallow
A–C	4,252	480	480	wrong label (2 distractors per instance)
A–D	2,126	240	240	literal paraphrase / no-evidence
B–D	1,702	192	191	literal paraphrase, vs. shallow preferred
A–E	959	117	109	over-interpretation
B–E	959	117	109	over-interpretation, vs. shallow preferred
Total	12,124	1,386	1,369	

Of the 959 training instances with a Tier E example, 422 are null on both axes, 535 are non-null instances whose gold intent is PP or AD, and 2 are NS with a substantive theme. The remaining 1,167 training instances, those whose gold intent is CI, PT, SI, ER, or RG, contribute no over-interpretation pair, which is the coverage gap noted above.

CGR threshold. The retrieval confidence threshold is $\theta = 0.55$. We characterize it on the validation set by encoding every instance's query and passage pool once and recording the maximum cosine similarity, which determines the gating decision: at $\theta = 0.55$ the gate withholds retrieval on 27.9% of validation instances, and Figure 4 reports the rate across $\theta \in [0.3, 0.9]$. The value is set from that distribution rather than by an accuracy sweep, which would require a full agent run per value and exceeded our compute budget. The distribution is what rules out lower values: the minimum observed maximum-similarity is 0.402 and the median is 0.599, so any $\theta \leq 0.40$, including the 0.3 we initially specified, gates 0% of instances and makes CGR a no-op, indistinguishable in the ablation from having no gate at all. We chose the smallest threshold that gates a non-trivial fraction, which keeps the gate conservative while leaving it functional. θ is *not* tuned for accuracy, and no configuration isolates the gate, so what is reported is the rate at which it fired on the test set (section 5.2) rather than an accuracy contribution.

Agent SFT. Supervised fine-tuning trajectories are synthesized from the gold labels, not demonstrated by human operators. For each of the 2,126 training instances, a fixed two-tool template is instantiated: an ANALYZERELATION call over the two speakers, an ANALYZETONE call on the target utterance, and a CONCLUDE call carrying the gold presence judgment, both gold labels, and the gold explanation, with the intervening thoughts and observations drawn from fixed strings. 5% is held out for validation, leaving 2,019 training trajectories. SFT runs for 2 epochs with a learning rate of 1×10^{-5} and batch size 16, with LoRA (rank 8, $\alpha = 32$, dropout 0.05) on the attention projections. The controller uses a smaller rank than the critic's 16; both are the defaults we started from and neither was tuned.

Two consequences follow and we state them rather than let the ablation carry an unexamined claim. The trajectories contain no variation in tool use, retrieval decisions, or stopping points, so SFT

teaches the output format and the label distribution, not the restraint policy, since every trajectory takes the same two actions regardless of whether the instance warrants depth. And because CONCLUDE always carries the gold label, SFT partly trains the controller directly on the reference labels, which is supervision the inference-time components do not receive. Any accuracy gain attributed to agent SFT should be read with that in mind; it is one of the reasons §7 notes that SFT adds supervision the ablations cannot fully disentangle from inference-time restraint.

Compute. All experiments were conducted on a single NVIDIA RTX 5090 (32GB), which is what makes LoRA and gradient accumulation necessary throughout rather than a design preference. BiCritic training takes 0.8 hours and agent SFT 0.2 hours, with retriever fine-tuning at 1.3 hours the largest of the three. Inference for the full RA-NarrAgent system on the 240-instance test set takes 11.6 hours at 173.9 seconds per instance, so evaluation costs roughly five times what training does.

C.3 FULL HYPERPARAMETER TABLE

Table 9 consolidates all hyperparameters across the three training stages and inference.

Table 9: Complete hyperparameter listing for all training stages and inference.

Component	Parameter	Value	Selection
Retriever (`bge-m3`, XLM-RoBERTa-large)			
	Learning rate	5×10^{-6}	Val. loss
	Batch size	4 ($\times$8 accum.)	GPU memory
	Epochs	3	Val. loss plateau
	InfoNCE temperature	0.07	Standard
	Warmup ratio	0.15	Standard
	Hard negatives per positive	3	Relation-aware mining
	Random negatives per positive	3	Standard
	Embedding dimension	1024	Model default
	Passage window size	3,000 chars	Dataset construction
	Window overlap	300 chars	Dataset construction
	Max passage length	128 tokens	Covers 300-char passages
	Max query length	256 tokens	Domain analysis
BiCritic (Qwen2.5-3B + LoRA)			
	LoRA rank	16	Not tuned
	LoRA α	32	$2 \times$ rank
	LoRA dropout	0.05	Standard
	LoRA target modules	All attention projections	Standard
	Learning rate	2×10^{-5}	Not tuned
	Warmup fraction	10%	Standard
	Schedule	Cosine decay	Standard
	Batch size	4 ($\times$4 accum.)	GPU memory
	Epochs	3	Val. loss plateau
	Direction loss weight λ	0.3	Not tuned
	BT margins, null	$A \succ B$ 0.15, $A \succ D$ 0.15, $A \succ C$ 0.75, $A \succ E$ 0.75, $B \succ E$ 0.50	
	BT margins, non-null	$A \succ B$ 0.25, $A \succ C$ 0.50, $A \succ D$ 1.00, $B \succ D$ 0.50, $A \succ E$ 0.75, $B \succ E$ 0.50	
	No C-vs-D or C-vs-E pair is formed in either condition. Two $A \succ C$ pairs are drawn per instance. Non-null E pairs are formed only for NS, PP and AD. Margins marked "domain analysis" were set by inspection of the tier definitions, not tuned on held-out data.		
Agent Inference			
	Max steps T	6	Not tuned; 3 is the *-Short* control
	CGR threshold θ	0.55	Gating-rate distribution
	Temperature	0.7 (0.3 for L3+NFP)	Not tuned
	Max new tokens	300	Fits observed answer lengths
	NFP evidence threshold	2 pieces	Domain analysis
	NFP confidence threshold	≤ 2	Domain analysis
	Maintain countdown	1	Fixed

C.4 CGR Gating Characterization

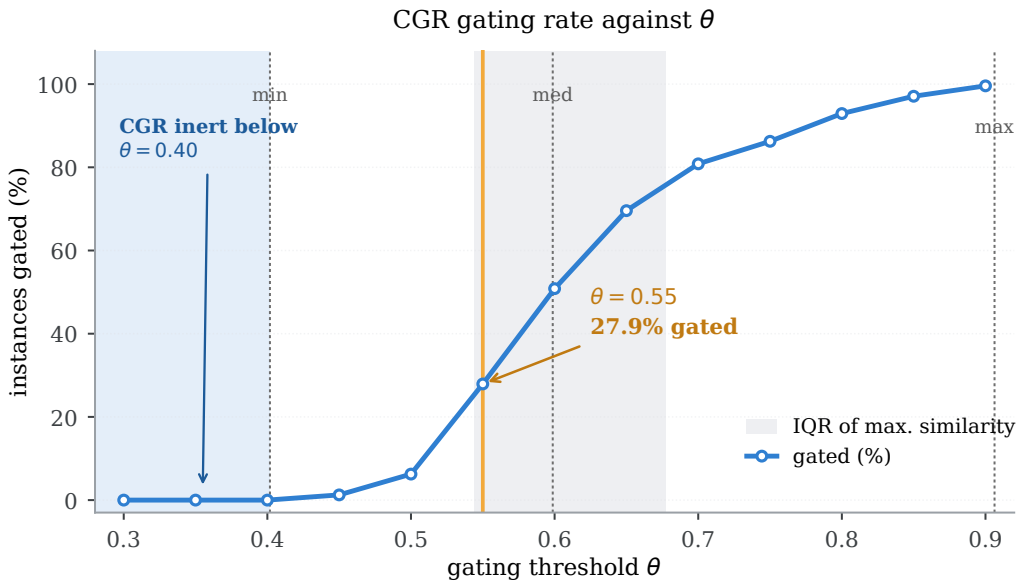


Figure 4: CGR gating rate against the threshold θ on the 240 validation instances, with the IQR of the observed maximum query-passage similarity shaded and its min, median and max marked. The gate is inert below $\theta = 0.40$, the minimum observed similarity, and near-total by 0.80, so the operating point $\theta = 0.55$ sits on the steep part of a distribution rather than on a tunable dial. The curve is the empirical CDF of maximum similarity, so it is exact at every θ from one encode pass.

We report how θ translates into gating behavior, not an accuracy-optimized sweep. Because the gate fires when the maximum query-passage cosine similarity falls below θ, one encode pass over the validation set yields the gating rate at every θ at once, and Figure 4 plots it over $\theta \in [0.3, 0.9]$ against the distribution of maximum similarities that produces it. The exact rates at the marked thresholds are 0.0% at 0.30 and 0.40, 1.2% at 0.45, 6.2% at 0.50, 27.9% at the operating point 0.55, 50.8% at 0.60, 80.8% at 0.70, 92.9% at 0.80, and 99.6% at 0.90. The maximum similarities themselves run min 0.402, p25 0.544, median 0.599, p75 0.677, max 0.906 over the 240 validation instances, and no instance has an empty passage pool.

Selecting θ by validation accuracy would require a full agent run at each of the thirteen values, which exceeded our compute budget. We therefore fix $\theta = 0.55$ from the similarity distribution alone. Three things follow for how the CGR result should be read. No configuration differs from RA-NarrAgent in the gate alone, so no accuracy contribution is attributed to CGR anywhere in this paper; what is reported is the rate at which it withheld a passage on the test set, 39 of 240 instances. Because θ was never fit to validation accuracy, the test-set numbers carry no threshold-selection overfitting. And the gating rate is the property that had to be checked before the gate could be said to exist at all: a threshold below the observed minimum similarity gates nothing, which is what the initially specified 0.3 did, making the mechanism a no-op under its own name.

Gating rate rises monotonically with θ by construction, and the mechanism the paper hypothesizes predicts that null-class recall rises with it while non-null instances lose useful context. We do not measure that tradeoff: doing so requires the per-θ agent runs just described. What Figure 4 establishes is only the input side of it, namely how often the gate fires at each threshold.

The compression of the similarities into roughly $[0.40, 0.91]$, with half the mass between 0.54 and 0.68, is what makes a gated dense retriever a sharp instrument on this pool rather than a smooth dial. Reporting the rate is what distinguishes a gate that withholds retrieval from one that is nominally enabled but never fires.

C.5 BiCritic Direction Loss Weight

The direction loss weight is fixed at $\lambda = 0.3$ and we report the critic at that operating point only: on the held-out preference pairs the scalar head ranks 93.4% of pairs correctly and the direction head reaches 95.2% three-way accuracy, with the A–C gap (gold versus wrong-label candidate) the hardest at 83.5%. The two heads share a backbone, so the natural question is whether they compete, that is whether buying direction accuracy costs ranking accuracy. We did not run the sweep that would answer it. Each additional λ needs a full critic retrain, and we chose to spend the available compute on the agent configurations and the two probes instead, so λ is an unswept hyperparameter and we make no claim that 0.3 is optimal. What the numbers above do establish is that at this setting neither head is starved: both clear 90% on their own objective, which is the property the joint loss was meant to preserve.

C.6 Reproducibility Checklist

We provide the following information to facilitate reproduction:

- **Code:** Full source code for all components (retriever, BiCritic, agent loop, evaluation metrics) will be released upon acceptance.
- **Data:** The LitSubText dataset (instances, annotations, and splits) will be released under a research-use license.
- **Models:** Trained retriever and BiCritic weights will be released.
- **Prompts:** All prompt templates are provided in Appendix F.
- **Random seeds:** All experiments use seed 42, fixed in advance. Each configuration is evaluated once; we do not report multi-seed variation (Appendix H).
- **Evaluation code:** The evaluation pipeline (including the null-passage stress test generator, consistency metrics, and grounding scorer) will be released as a standalone package.
- **Hardware:** All experiments were conducted on a single NVIDIA RTX 5090 (32GB). The full experimental pipeline, training plus inference for every configuration on both backbones, required 34 GPU-hours, of which 2.3 are training.

D Full Per-Class Results

Table 11 presents complete per-class precision, recall and F1 for intent classification across key systems, and Figure 3 plots the intent F1 comparison referenced in section 6. Table 10 gives the cross-model replication of section 5.5.

Table 10: Cross-model replication. Macro-F1 (%) on both label axes for L0 (zero-shot), L3 (decomposition peak), NarrAgent (full agent), and RA-NarrAgent across two LLM backbones from different families. Only the controller changes; the critic and retriever are the same trained components in both rows. The two rows are *not* comparable at the level of absolute values: the controller LoRA is shape-bound to the Qwen backbone, so the Llama row runs an untrained controller (section 5.5). What replicates, or fails to, is the within-row ordering, whether the peak sits at a prompt-only configuration and whether the agent loop costs accuracy. The final column reports the rate (%) at which the loop exhausted its six-step budget without emitting a conclusion, for NarrAgent and for RA-NarrAgent respectively; this is the mechanism behind the Llama agent columns and is not a scoring artefact (Appendix E). That RA-NarrAgent's non-termination rate on Llama is 10.0% against NarrAgent's 60.8% is consistent with the IDC doing what it was designed to do, forcing a conclusion, on a controller that was never trained to conclude on its own.

		L0		**L3**		**NarrAgent**		**RA-NarrAgent**		**No concl.**
Controller	**Adapter**	Int.	Thm.	Int.	Thm.	Int.	Thm.	Int.	Thm.	NA / RA-NarrAgent
Qwen2.5-7B-Instruct	SFT+DPO	17.9	14.1	18.2	27.3	8.8	4.9	15.5	5.7	0.0 / 0.0
Llama-3.1-8B-Instruct	none	7.3	17.7	14.8	30.2	3.6	3.7	12.3	12.7	60.8 / 10.0

Intent class analysis. Three patterns hold across the per-class intent results. The first is category attractiveness. NarrAgent recovers the three analytically attractive classes (PT, CI, SI) at a mean recall of 20.6% against 0.7% on the three restrained classes (AD, RG, NS), a gap of 20.0 points,

Table 11: Full per-class intent classification results (Precision / Recall / F1, %) for L3, NarrAgent, and RA-NarrAgent on the LitSubText test set.

	L3			NarrAgent			RA-NarrAgent		
Class	P	R	F1	P	R	F1	P	R	F1
PT	18.6	52.4	27.5	11.0	38.1	17.0	22.7	23.8	23.3
SI	8.6	63.6	15.2	0.0	0.0	0.0	20.0	9.1	12.5
CI	30.2	30.2	30.2	28.8	23.8	26.1	28.6	28.6	28.6
ER	42.9	12.5	19.4	12.9	45.8	20.2	11.8	50.0	19.0
PP	57.9	20.8	30.6	14.3	1.9	3.3	50.0	7.5	13.1
AD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RG	100.0	8.3	15.4	0.0	0.0	0.0	0.0	0.0	0.0
NS	50.0	4.1	7.5	50.0	2.0	3.9	52.9	18.4	27.3
Macro	38.5	24.0	18.2	14.6	14.0	8.8	23.3	17.2	15.5

and L3 shows the same asymmetry more sharply at 48.7% against 4.1%, a gap of 44.6 points. RA-NarrAgent narrows the gap to 14.4 points, but the narrowing comes almost entirely from the restrained side, which rises from 0.7% to 6.1%, while attractive-class recall is unchanged at 20.5%. The mechanisms therefore recover a little more of the classes that require holding back rather than suppressing the categories a reader is drawn to.

The second is that precision on the attractive classes stays low everywhere. NarrAgent reaches 11.0% on PT and 28.8% on CI, so most predictions of these labels are wrong. RA-NarrAgent roughly doubles PT precision to 22.7% and leaves CI unchanged at 28.6%, which is a smaller effect than the restraint mechanisms were designed to produce. Prediction mass moves rather than concentrating on the correct instances: RA-NarrAgent emits ER 102 times out of 240 against 24 gold ER instances, at 11.8% precision, so the configuration that is most restrained about PT is simultaneously the most committed to a different non-null class.

The third is that AD and RG are the hardest classes for every system. AD reaches 0.0 F1 under all three configurations and RG is nonzero only for L3, at 15.4. Both describe communicative functions, deflection and relationship management, that no single surface cue marks, and with 7 and 12 gold instances respectively neither class has enough support for these figures to be read as anything more than an upper bound on how often a system lands on them at all.

Theme class analysis. The theme axis varies more across systems than the intent axis, not less, and every agent configuration loses more ground there. Against L3, theme macro-F1 falls by 14.3 points for NarrAgent+NFP, 22.4 for NarrAgent and 21.6 for RA-NarrAgent, while the corresponding intent losses are 4.8, 9.4 and 2.7. The asymmetry runs opposite to what we expected when designing the analysis: whatever the loop does to predictions, it does more of it to the abstract thematic categories than to the specific communicative functions. The most difficult theme classes for all systems are MH (Moral Hypocrisy) and DN (Desire vs. Norm), both of which require recognising a thematic pattern that no single lexical cue signals.

The two null classes behave alike, and both show that the null-first prompt, not restraint, is what moves them. L3 almost never predicts either null class, with 4.1% recall on NS and 4.2% on NT out of 49 and 48 gold instances respectively, and NarrAgent is lower still at 2.0% and 0.0%. Adding the null-first prompt raises NarrAgent's NS recall to 22.4% and NT recall to 14.6%. RA-NarrAgent, which contains that prompt as a component, reaches 18.4% and 10.4%, below NarrAgent+NFP on both. So the null-class gain is attributable to the prompt, and the restraint mechanisms layered on top of it recover slightly less of the null classes rather than more. This is the same pattern section 6 reports on the intent axis, and it is the reason the theme comparison between RA-NarrAgent and NarrAgent+NFP goes against RA-NarrAgent.

Cross-axis correlations. Whether correctness on one axis predicts correctness on the other depends on the system, and the pattern does not support the reading we anticipated. For L3 the two axes are essentially independent: of 240 instances, 20 are correct on both, 35 on intent only, 62 on theme only and 123 on neither, giving $\chi^2 = 0.2$, $p = 0.70$, $\phi = +0.03$. The agent configurations all show a positive association, $\phi = +0.18$ for NarrAgent ($\chi^2 = 7.7$, $p = 0.005$), $+0.19$ for NarrA-

gent+NFP ($\chi^2 = 8.8$, $p = 0.003$) and $+0.28$ for RA-NarrAgent ($\chi^2 = 18.7$, $p < 10^{-4}$), but a rising ϕ here is not evidence of improved coherence between the axes. It is largely what happens when one axis collapses: RA-NarrAgent is correct on theme for only 16 instances, so its theme correctness is close to constant and co-varies with almost anything. Reading ϕ as coherence would credit a system for the narrowness of its theme predictions.

The direction of the partially-correct cases is more informative. For L3, 63.9% of them are correct on theme but wrong on intent; for the agent configurations that flips, reaching 86.7% correct-on-intent-but-wrong-on-theme for RA-NarrAgent. The loop does not distribute its errors evenly across the two axes, and shifts them onto theme, which is consistent with the axis-asymmetric degradation reported above.

E Two Distinct Failure Modes in the Llama Row

The Llama configurations initially reported parse failure rates between 62.9% and 84.2%, which would make their scores uninterpretable. Inspecting the persisted raw generations showed two unrelated causes, one of which is a defect in our scoring and one of which is a property of the system being measured. We separate them here because they have opposite implications: the first had to be repaired before the row meant anything, and the second must not be.

A scoring defect in the prompt-only configurations. Llama-3.1-8B-Instruct emits the label codes as bare identifiers inside the JSON object it is asked for, as in {`"subtext_presence": true, "subtext_intent": CI, "theme_reaction": EM, ...`}. Bare `CI` is not a valid JSON token, so the object failed to parse, the prediction was stored empty, and the instance was scored as wrong. The model's answer was in fact present and well-formed apart from the missing quotation marks. We recover it offline: a bare two-character token in value position is quoted, and only when it is a legal code for that axis. We do not mine the surrounding prose for codes, since doing so would manufacture a prediction for instances where the model never committed to one, which is the failure mode our label parser was written to prevent. Of 565 affected baseline instances 564 are recovered, and parse failure falls from 72.1–84.2% to 0.0–1.7%. The generations were persisted precisely so a parser fix could be replayed without re-running the model, and no GPU time was needed. Applying the same repair to the Qwen runs changes 19 instances and moves no reported figure by more than 0.8 points, so the main results are unaffected; we report the repaired numbers for both backbones.

Non-termination in the agent configurations. The Llama agent rows are a different matter. On 146 of 240 instances the recorded failure is `no conclusion within budget`: the controller consumed its six-step budget cycling through `Verify` and `Hypothesize` calls and never emitted the `Conclude` action that produces an answer. Nothing is recoverable here, and re-parsing recovers nothing, correctly, because there is no answer in the transcript to recover. This is the untrained controller failing to terminate, and at the same step budget the SFT'd Qwen controller does so on 0.0% of instances against Llama's 60.8%. We therefore report it as a non-termination rate in Table 10 rather than treating it as a measurement error. It does mean the Llama agent columns are low for a coarser reason than the one this paper is about: they reflect a loop that does not finish, not a loop that over-interprets.

F Prompt Templates

This appendix reproduces the prompt templates as they appear in the code that produced the reported runs, with the placeholders {`dialogue`} for the formatted dialogue excerpt, {`scene_context`} for the stage directions, {`target_utterance`} for the focal utterance and {`retrieved_passages`} for the top-3 retrieval results. All four prompt-only configurations end with the same output specification, reproduced once below, so the configurations differ only in the context they prepend and in whether they add the decomposition template.

Shared output specification. Appended verbatim to every prompt-only configuration.

```
You must output a JSON object with exactly these keys:
{"subtext_presence": true or false, "subtext_intent": one
of [PT, ER, CI, SI, PP, AD, RG, NS], "theme_reaction": one
of [CC, GC, IA, FO, EM, SE, MH, DN, NT], "explanation":
"2-3 sentence explanation of the hidden meaning with textual
evidence"}
```

`Subtext-intent options:` followed by the eight intent codes with their names, then `Theme-reaction options:` followed by the nine theme codes with theirs. Both lists are generated from the label dictionaries in the code, so the names given to the model are exactly those of Table 3 and of the theme list in section 3.1.

Shared L0 through L2 prompt with incremental blocks. Each level prepends one block to the previous level and rewrites only the closing instruction, so the three are given together with the additions bracketed by the level that introduces them.

```
You are an expert analyst of subtext in film dialogue -
hidden meaning conveyed through implication, irony, or
socially constrained expression.
```

```
[L1 adds] SCENE CONTEXT (stage directions / action lines):
{scene_context}
```

```
[L2 adds] RETRIEVED PASSAGES FROM THE SCREENPLAY:
[1] (chunk_71, relevance 0.60) {passage text} ...
```
(top-3, each with its chunk reference and retriever score)

```
DIALOGUE: {dialogue}
TARGET UTTERANCE: "{target_utterance}"
```

```
Analyse this dialogue for subtext.
```
(L0)
```
Using the scene context to inform your reading, analyse this
dialogue for subtext.
```
(L1)
```
Use the scene context and retrieved passages to restore
narrative background before analysing the dialogue for
subtext.
```
(L2)

Every level then appends the shared output specification above.

L3 adds the six-field decomposition template. L3 keeps the L2 context unchanged and replaces the closing instruction with a two-step template over six named fields. This is the configuration that attains the best theme macro-F1 in Table 1, so its exact wording matters.

```
[L2 role sentence, scene context, retrieved passages,
dialogue, target utterance]
```

```
Step 1 - Structured subtext decomposition. Before
predicting labels, reason through each dimension:
m_lit   (literal meaning): What is the surface meaning of
the target utterance?
m_emo   (emotional stance): What emotion is expressed or
suppressed?
m_rel   (relation dynamics): How does this reflect the
relationship between the speakers?
m_conf  (latent conflict): What underlying tension is
present?
m_sub   (subtext intent): What interpersonal function does
the hidden meaning serve?
m_theme (thematic reaction): What broader socio-literary
tension does this reflect?
```

```
Step 2 - Output your final predictions.
```
[shared output specification, with "You must output" replaced by "For Step 2, output"]

```
Respond with a JSON object that includes both a
"decomposition" key (with the six fields above) and the
four prediction keys.
```

Configurations that were planned but not run. Three configurations named in earlier drafts have no prompt here because they were never executed: majority-vote self-consistency, critic reranking over sampled candidates, and the external agent frameworks. Each would have multiplied an inference budget already at 174 seconds per instance, and none is reported in any table.

NarrAgent agent prompt without the null-first block. Every agent configuration builds its prompt from one function, and the configurations differ only in whether the null-first block is inserted. Without it the prompt reads as follows, with the tool list reproduced verbatim.

```
You are NarrAgent, an expert literary subtext analyst.
Analyze the following dialogue for hidden meaning (subtext).

Available tools:
- Retrieve(query): Search for relevant passages from the
screenplay. Returns top-3 passages.
- AnalyzeRelation(char_a, char_b): Analyze the
power/emotional dynamics between two characters.
- AnalyzeTone(utterance): Analyze the tone and emotional
register of an utterance.
- Hypothesize(context): Generate 2-3 candidate subtext
interpretations based on gathered evidence.
- Verify(hypothesis): Score a hypothesis using the
interpretation critic. Returns a quality score.
- Conclude(subtext_presence, subtext_intent, theme_reaction,
explanation): Output final prediction. Must be the last
action.

SCENE CONTEXT: {local_context}
DIALOGUE: {dialogue}
TARGET UTTERANCE: "{target_utterance}"

Analyze this step by step. For each step, output:
Thought: <your reasoning>
Action: <tool_name>(<args>)

When ready, call Conclude with EXACT codes only:
subtext_intent must be one of: PT/ER/CI/SI/PP/AD/RG/NS
theme_reaction must be one of: CC/GC/IA/FO/EM/SE/MH/DN/NT
confidence must be an integer 1-5 (how confident you are
that subtext exists)
Example: Conclude(subtext_presence=<true|false>,
subtext_intent=<CODE>, theme_reaction=<CODE>,
confidence=<1-5>, explanation=<one or two sentences>)
```

The call-shape example uses placeholders rather than concrete codes, and this detail is load-bearing. An earlier version of the prompt wrote two real labels into the example, and the controller copied them instead of deciding, which concentrated theme predictions on one class across every agent configuration while the prompt-only rows, whose prompts come from a different builder and carry no example, were unaffected. All agent runs reported here use the placeholder form, so the residual class concentration reported in section 6 is a property of the model rather than of the example.

The null-first block. RA-NarrAgent, NarrAgent+NFP and NarrAgent+Scalar insert the following block immediately after the role sentence of the prompt above, leaving the tool list, the context fields and the call specification unchanged. L3+NFP appends the same text to the prompt-only L3 prompt.

```
IMPORTANT --- Null-Hypothesis Protocol:
Begin by assuming NO subtext exists (null hypothesis). Only
revise this assumption if you find at least TWO independent
pieces of positive evidence from the dialogue itself. If
evidence is ambiguous or based solely on retrieved dramatic
context, maintain the null hypothesis. Before choosing a
label, rate your confidence that subtext exists on a 1-5
scale. If confidence <= 2, output NS for intent and NT for
theme.
```

The last sentence is enforced outside the model rather than trusted to it. Any CONCLUDE carrying a confidence of 1 or 2 is rewritten to NS on intent and NT on theme whatever labels it generated, and the rewrite also clears a label-parse failure on that instance, since a deliberate null supersedes an unresolvable code. The rewrite sits in the output parser, so it is armed for every agent configuration, including the two that do not carry the block above, because the call specification asks for a confidence value regardless. What differs across configurations is how often the model lands under the threshold. It fires on 4 of 240 instances for RA-NarrAgent, 12 for NarrAgent+NFP, 16 for NarrAgent+Scalar, 5 for L3+NFP, 1 for NarrAgent-Short and 0 for NarrAgent.

External frameworks. NarrAgent is itself a ReAct loop over task-specific tools on the same backbone, so the comparison this paper's claim requires is between restrained and unrestrained depth inside one loop rather than against a differently-scaffolded system. ReAct, Reflexion and Self-RAG are accordingly treated as related work in section 2 and no adapted prompt exists for them.

BiCritic input format. The critic receives a system message and one user message assembled from the instance and the candidate interpretation.

```
System: You are a literary analysis critic. Evaluate the
quality of the following subtext interpretation.

DIALOGUE: {dialogue}
TARGET UTTERANCE: "{target_utterance}"
SCENE CONTEXT: {local_context}
PROPOSED INTERPRETATION:
Subtext intent: {code} ({name})
Theme reaction: {code} ({name})
Explanation: {explanation}
Evidence: {spans joined by semicolons, or [no evidence
cited]}
```

Two details of this format matter for reading the probe results of Appendix G. The label fields carry both the code and its full name, so a candidate that differs from another only in its label differs in a short, fixed region of the input, which is what makes the critic probe's paired comparison meaningful. And inside the loop the interpretation handed to VERIFY is the argument string of the VERIFY call, from which intent and theme codes are recovered by string match and default to NS and NT when neither code appears, so a trajectory that verifies a prose hypothesis without naming a label is scored as a null candidate. The critic reads a scalar reward and a three-way direction over DEEPEN, MAINTAIN and SIMPLIFY from the final hidden state, and the direction passed back to the controller is the argmax of that head.

G CRITIC AND CONTROL PROBES

This appendix specifies the two probes of section 5.4–5.4. Both hold the system fixed and vary a single factor, so neither depends on the multi-variable ablation rows of Table 2.

G.1 CRITIC PROBE WITH A FIXED INPUT AND VARIED CANDIDATES

Confounds excluded by the probe. BiCritic receives the dialogue, the scene, and a candidate interpretation. A critic that ignored the candidate could still score well on tier-labelled data by learning a shortcut from the input alone: passages that look null-ish attract SIMPLIFY, analytically rich passages attract DEEPEN. Such a critic would not be locating unsupported claims, and the IDC built on it would be an input-conditioned length prior rather than a restraint mechanism. The probe fixes the input and varies only the candidate, so any direction difference must come from the candidate.

Candidate construction. We sample 109 validation instances stratified by gold null status (half NS, half non-NS on the intent axis) and, for each, construct four candidates in a 2×2 design crossing *correctness* (the gold intent label vs. a distractor label) with *depth* (concise vs. detailed):

- **Correct/concise:** the gold label with one licensed evidence span and no further inference.

- **Correct/detailed:** the same gold label plus two added claims that the passage does not license, namely inferred backstory, unstated motives and symbolic readings of incidental detail. The label is right; the claim set is inflated.
- **Wrong/concise:** a distractor label with thin support, i.e. an error of under-interpretation rather than excess.
- **Wrong/detailed:** the distractor label with the same kind of added unlicensed claims.

Distractor labels are drawn from the confusion neighbours of the gold class, the adjacent pairs PT/PP, ER/AD and SI/RG, rather than uniformly, so wrong candidates are plausible. The "detailed" candidates are the concise ones with two fixed sentences appended, written to assert content no passage can establish, such as an earlier humiliation, a decision predating the conversation, the setting mirroring an inner state, or an unspoken bargain. Using a fixed, label-independent set rather than per-instance generation is what makes the 2×2 clean: the same two sentences are added in every cell, so "detailed" differs from "concise" by exactly those sentences and nothing else, and the additions cannot be tailored to the instance. The cost is that the additions are generic and will occasionally be closer to the passage than intended; they are not screened per instance, so the manipulation is over-assertion in the average case rather than guaranteed to be unlicensed on every instance. Candidate length is recorded so a direction difference can be checked against a length-only account, which matters here because appending fixed text also makes "detailed" uniformly longer.

Baselines. Two reference critics bound the result. The *empty-candidate* critic receives the same input with the candidate field blank, giving the direction distribution obtainable from the input alone, the blind floor. The *scalar* critic is NarrAgent+Scalar, which has no direction head; we threshold its reward to induce a two-way signal for comparison.

Analysis. All four candidates for an instance are scored by the same critic checkpoint, so comparisons are within-instance and paired. We report the SIMPLIFY rate for each cell of the 2×2, and test two orderings with paired tests over instances: detailed $>$ concise at matched correctness (the excess-detection claim), and the same contrast on correct candidates alone (the stronger claim, since there the label is right and only the claim set is inflated). A candidate-blind critic yields equal rates in every cell; the empty-candidate rate quantifies how much of the observed rate is input-driven. Table 12 reports the cells with within-instance differences and confidence intervals.

Table 12: Critic probe. SIMPLIFY rate (%) by candidate correctness and depth, same input within each instance. Δ_{depth} is the paired detailed$-$concise difference with a 95% CI over instances. The empty-candidate row is the blind floor: a candidate-blind critic would give identical rates across columns.

Critic / candidate set	Concise	Detailed	Δ_{depth} (95% CI)
BiCritic, correct label	4.6	12.8	$+8.3$ $[+5.8, +10.3]$
BiCritic, wrong label	50.5	51.4	$+0.9$ $[-5.2, +6.7]$
Empty candidate (blind floor)	0.0		—

$n = 109$ instances, 12 screenplays; CIs from a screenplay-level cluster bootstrap.
Correctness contrast on concise candidates: $+45.9$ $[+41.4, +50.9]$, $p < 0.001$.
Mean critic score by cell: $+4.21$ / $+1.87$ (correct), -3.80 / -4.06 (wrong), -10.76 (empty).

Three things follow. The blind floor is 0.0%: with the candidate field blank the critic emits DEEPEN on all 109 instances and SIMPLIFY on none, so the SIMPLIFY signal cannot be produced from the input alone and the shortcut account is ruled out. The depth effect on correct candidates is small but reliably positive, $+8.3$ points with a CI excluding zero, so the critic does detect inflated claim sets when the label is right, which is the excess-detection claim in its cleanest form. And the depth effect vanishes on wrong candidates ($+0.9$, CI spanning zero) because the correctness signal has already saturated the response: wrong candidates draw SIMPLIFY half the time regardless of how much unlicensed detail is added. The correctness contrast is 5.5 times the depth contrast on concise candidates, $+45.9$ against $+8.3$ points, so the critic is substantially a correctness detector with a secondary sensitivity to over-assertion, not primarily an over-assertion detector. We did not run the thresholded scalar-critic comparison: NarrAgent+Scalar has no direction head, and choosing the

threshold that converts its reward into a two-way signal is itself a free parameter with no principled setting, so the comparison would report our threshold choice rather than a property of the critic.

G.2 Control Probe Contrasting Advice With Enforcement

Quantity isolated by the probe. The IDC both supplies the direction signal and enforces it. Removing the IDC therefore removes information and control at once, which is why the ablation cannot separate them. Here the critic, its checkpoint, and the feedback string are held identical, and only the control regime changes: under *advisory*, the signal is appended to the history and the agent may act on it or continue; under *enforced*, SIMPLIFY triggers the revision pass and ends the loop as in Algorithm 1. Any difference between the two is attributable to enforcement rather than to what the critic knows.

Measurements. Beyond accuracy, FPR, and Harm, we log the *override rate*: how often the advisory agent takes a further interpretive step after receiving SIMPLIFY, and how often that step adds claims rather than revising. A high override rate with unchanged accuracy would mean the signal was ignored harmlessly; a high override rate with degraded accuracy is direct evidence that enforcement, not information, carries the effect. Mean steps are reported to confirm the two regimes are not simply different budgets.

Table 13: Control probe. Same critic and feedback under advisory vs. enforced control, $n = 60$ instances per arm over 12 screenplays. *Simp.* is the number of SIMPLIFY signals the critic emitted; the override rate is undefined when it is zero, which is what happened in both arms. The accuracy difference is not significant (95% CI $[-3.2, +19.0]$, $p = 0.208$).

Regime	Int. Acc.	FPR ↓	Mean steps	Simp.	Override
Advisory	15.0	27.8	3.77	0	n/a
Enforced (IDC)	23.3	20.5	3.87	0	n/a
Δ (enforced − advisory)	$+8.3$	-7.3	$+0.10$	—	—

Parse failure 1.7% in both arms. FPR scored on 36 / 39 null-probe instances respectively.
Critic directions, advisory: 15 MAINTAIN, 9 DEEPEN; enforced: 13 MAINTAIN, 9 DEEPEN.

Outcome and its shortfall against the design. Enforcement is ahead on both metrics, $+8.3$ points of intent accuracy and 7.3 points less FPR, but the accuracy interval spans zero ($[-3.2, +19.0]$, $p = 0.208$) at $n = 60$ per arm, so the probe does not establish that enforcement matters. Mean steps are nearly identical (3.77 vs. 3.87), which at least confirms the two regimes are not different budgets in disguise.

The design assumption that failed is the more informative result. The probe measures what happens when an agent is allowed to override SIMPLIFY, and the critic emitted SIMPLIFY zero times in either arm on these 60 instances, since every direction was MAINTAIN or DEEPEN. The advisory agent therefore never had a signal to ignore, and the two arms are closer to two runs of the same system than to a contrast. This is consistent with Appendix G.1: SIMPLIFY requires a wrong or over-detailed candidate, and an agent's own in-loop draft on a typical instance is neither in a way the critic detects. A probe that actually tests enforcement needs instances pre-selected for eliciting SIMPLIFY, which we did not do here; we report the null rather than reframe it.

H Statistical Tests

All reported differences between RA-NarrAgent and baselines are tested for statistical significance using the following procedures.

Tests and the single-run design. Per-instance correctness is compared with McNemar's test for paired nominal data, with Cohen's g as the effect size; macro-F1 differences use a bootstrap of 10^4 resamples reported as 95% confidence intervals. Harm and Repair are per-instance binaries against a fixed set of seed-42 L3 predictions, so every system is measured against the same reference, and they are tested the same way.

Each configuration was evaluated once, at seed 42, fixed in advance as the project default (Appendix C). There are no multi-seed standard deviations to report: with nine main-table configurations, three cross-model ones, five ablations and two probes, a three-seed protocol would have tripled a budget that one pass already consumed over some thirty GPU-hours on a single device. Two consequences bound every claim. Run-to-run variance under stochastic decoding is unmeasured, so we cannot say whether a gap of two or three points between adjacent configurations would survive a re-run, and we accordingly draw conclusions only from the large orderings, prompt-only above every agent configuration by several points on both axes, treating adjacent agent configurations as indistinguishable. And every test below operates on sampling variation with the predictions held fixed, so an interval answers whether a gap would persist on another 240 instances from this population, not whether it would persist on another run of this system. The design does make the tests genuinely paired, since each instance contributes exactly one prediction per system, and there is no seed-selection step through which an outcome could influence which predictions are tested.

Consistency identity as an internal check. Under a fixed L3 reference on a fixed dataset, accuracy decomposes exactly:

$$A_{\text{full}} = A_0(1 - \text{Harm}) + (1 - A_0)\,\text{Repair}, \tag{1}$$

where A_0 is L3 accuracy, 22.5% on intent and 34.2% on theme, and A_{full} is the accuracy of the system under comparison. We verify Eq. 1 for every row of Table 1 as an internal consistency check; any row where the reconstructed accuracy differs from the reported accuracy by more than floating-point tolerance indicates a bookkeeping error, such as Harm and Repair computed against a different L3 run than the one reported. We recommend the check to readers reproducing our numbers.

The identity constrains the arithmetic but does not license a causal reading. Harm and Repair explain *where* an accuracy change comes from, in the sense of which subset of instances moved, and not *why* it happened. In particular the configurations compared here differ simultaneously in training (agent SFT), prompting (NFP), retrieval (CGR gating), and tool invocation, so a low Harm rate is evidence that a system preserves the reference's correct predictions, not evidence that inference-time restraint specifically caused the preservation. Attributing the change to reasoning depth alone would require holding the other three dimensions fixed, which our ablations approximate but do not achieve; §7 states this confound.

Full significance table. Table 14 reports the pairwise significance results for all comparisons discussed in the paper, separating the pre-specified confirmatory family from the exploratory comparisons. Holm-adjusted p-values are reported only for the confirmatory family, since applying a familywise correction to exploratory comparisons would imply they carry confirmatory weight. All entries are computed from the single seed-42 run per system defined above.

Ablation significance. For the ablation study (Table 2), we test each partial ablation against the full RA-NarrAgent system. These are exploratory comparisons in the sense defined below, reported with uncorrected p-values. We had pre-registered FPR and a harm rate as the IDC's primary outcomes and accuracy as secondary, on the reasoning that early stopping acts on over-commitment rather than on correctness. Only the second of those is available: the null-passage stress set was never run, so the pre-registered FPR outcome has no measurement outside the control probe's two arms, and we report accuracy and macro-F1 as the IDC's outcomes instead. We flag the substitution rather than presenting accuracy as the original plan.

Harm and repair rates. For a pair of systems $A \to B$ we also report Harm $= P(B \text{ wrong} \mid A \text{ correct})$ and Repair $= P(B \text{ correct} \mid A \text{ wrong})$, which decompose an accuracy difference into what a change breaks and what it fixes. The decomposition is informative precisely where the net difference is small: L3+NFP versus L3 nets $+2.9$ points of intent accuracy, but reaches it by breaking 92.6% of what L3 got right while repairing 30.6% of what L3 got wrong. On the theme axis the same substitution harms 89.0% and repairs 3.2%. A net change of this shape is not a partial improvement; it is a different predictor.

Power analysis. McNemar's test depends only on the discordant pairs, so sample size and baseline accuracy are not sufficient to determine power: the operative quantity is the discordance proportion

Table 14: Pairwise significance tests. Accuracy comparisons use McNemar's χ^2 (with continuity correction) and Cohen's g; macro-F1 comparisons use a screenplay-level cluster bootstrap (10^4 resamples of the 12 screenplays) reported as Δ with a 95% CI, because macro-F1 is not an instance-level mean and cannot be tested pairwise per instance. p_{Holm} is the Holm–Bonferroni-adjusted value within the confirmatory family; "—" marks exploratory comparisons, which are not corrected and are not confirmatory evidence. We report no corpus-level FPR or evidence-grounding test: the null-passage stress set was not run, so FPR exists only for the two control-probe arms (Table 13), and the grounding metric is unvalidated (section 6).

Metric	Comparison	Stat.	p	p_{Holm}	Eff.	95% CI
Confirmatory family (pre-specified)						
Int. acc.	RA-NarrAgent vs. NarrAgent+NFP	$\chi^2 = 0.1$	0.716	1.000	$g = 0.03$	
Int. acc.	RA-NarrAgent vs. L3	$\chi^2 = 0.2$	0.640	1.000	$g = 0.03$	
Int. acc.	RA-NarrAgent vs. L3+NFP	$\chi^2 = 1.6$	0.201	0.804	$g = 0.08$	
Int. acc.	RA-NarrAgent vs. NarrAgent+Scalar	$\chi^2 = 0.4$	0.544	1.000	$g = 0.04$	
Thm. acc.	RA-NarrAgent vs. NarrAgent+NFP	$\chi^2 = 3.0$	0.082	0.246	$g = 0.15$	
Thm. acc.	RA-NarrAgent vs. L3	$\chi^2 = 50.3$	< 0.001	< 0.001	$g = 0.39$	
Exploratory (uncorrected)						
Int. acc.	RA-NarrAgent vs. NarrAgent-Short	$\chi^2 = 1.8$	0.178	—	$g = 0.20$	
Int. acc.	RA-NarrAgent vs. NarrAgent	$\chi^2 = 2.7$	0.099	—	$g = 0.25$	
Thm. acc.	RA-NarrAgent vs. NarrAgent	$\chi^2 = 0.1$	0.728	—	$g = 0.09$	
Intent acc.	L3+NFP vs. L3	$\chi^2 = 0.3$	0.562	—	$g = 0.03$	
Theme acc.	L3+NFP vs. L3	$\chi^2 = 57.6$	< 0.001	—	$g = 0.44$	
Intent acc.	NarrAgent+NFP vs. L3	$\chi^2 = 0.7$	0.391	—	$g = 0.05$	
Theme acc.	NarrAgent+NFP vs. L3	$\chi^2 = 30.5$	< 0.001	—	$g = 0.29$	
Int. macro-F1	NarrAgent+NFP vs. L3	$\Delta = -4.8$	0.177	—		$[-11.2, +2.3]$
Thm. macro-F1	NarrAgent+NFP vs. L3	$\Delta = -14.4$	0.007	—		$[-22.7, -3.8]$
Int. macro-F1	RA-NarrAgent vs. NarrAgent+NFP	$\Delta = +2.1$	0.422	—		$[-3.9, +7.5]$
Int. macro-F1	RA-NarrAgent vs. L3	$\Delta = -2.7$	0.401	—		$[-8.2, +3.2]$
Int. macro-F1	L3+NFP vs. L3	$\Delta = -6.7$	0.015	—		$[-11.1, -1.2]$
Thm. macro-F1	L3+NFP vs. L3	$\Delta = -22.7$	< 0.001	—		$[-27.0, -16.3]$

π_d, the fraction of instances on which the two systems disagree. Writing Δ for the accuracy difference, the test statistic is approximately $\chi^2 = n\Delta^2/\pi_d$, so at fixed n and Δ power *decreases* as π_d grows, and the minimum detectable effect scales as $\sqrt{\pi_d/n}$. The intuition is that a net difference of a given size is easier to distinguish from noise when it arises from few disagreements than when it is the residue of many disagreements in both directions. Because $\pi_d \geq |\Delta|$, low discordance and a large effect cannot co-occur without bound; the informative case is two systems that differ on many instances while netting out to little, which is the low-power regime. We therefore state π_d explicitly rather than deriving power from n and A_0 alone.

With $n = 240$ test instances and an observed discordance of $\pi_d = 0.304$ between RA-NarrAgent and L3 on the intent axis, we have 80% power at $\alpha = 0.05$ (two-sided) to detect an absolute accuracy difference of 10.0 percentage points. The dependence on π_d is material: holding n fixed, the minimum detectable effect at $\pi_d = 0.40$ is twice that at $\pi_d = 0.10$. Discordance across the five confirmatory comparisons runs from $\pi_d = 0.221$ against NarrAgent to 0.308 against L3+NFP, so the minimum detectable effect varies only between 8.5 and 10.0 points and no comparison in the family is materially better powered than another.

The observed discordance is high in practice, which is the unfavourable regime: L3+NFP versus L3 disagrees on 44.6% of instances on the intent axis while netting 2.9 points, giving an MDE of 12.1 points, four times the observed difference. Two systems can therefore differ substantially in behaviour and still produce an accuracy comparison this test cannot resolve, which is why we read the harm/repair decomposition and the per-class breakdown alongside it rather than the net alone.

For per-class comparisons, power is more limited still. The smallest intent class in the test set contains 7 instances, so per-class differences are reported as descriptive and we base claims on aggregate patterns, the gap between attractive and restrained classes, rather than on individual class comparisons.

Multiple comparisons. Correction requires a comparison family fixed in advance, otherwise the divisor depends on how many tests we chose to report. We pre-specify the confirmatory family as four system pairs bearing on the paper's claims: RA-NarrAgent against NarrAgent+NFP (the recovery claim), against L3 and against L3+NFP (the claim that any gain is not prompting alone), and against NarrAgent+Scalar (the claim that the direction head contributes beyond a scalar critic). Each pair is tested on both label axes, and Holm–Bonferroni is applied within each axis over the four pairs, so the correction factor is four rather than eight: the two axes are separate outcome families, and pooling them would let a result on one axis raise the bar for the other. The pre-registered version of this family also included an FPR outcome; since the null-passage set was not run, that member has no measurement and is dropped rather than silently replaced, leaving the four pairs above. We apply Holm–Bonferroni within the family, which controls the familywise error rate while being uniformly more powerful than plain Bonferroni. All remaining comparisons, the per-component ablations in Table 2, the cross-model results in Table 10 and all per-class breakdowns, are exploratory: we report uncorrected p-values and do not treat them as confirmatory evidence.

Table 14 reports uncorrected and Holm-adjusted p-values for the full confirmatory family. Both the family membership and the ordering of outcomes above were fixed before correction was applied, and every member is reported in that table whether or not it reaches significance.

Bootstrap details. For macro-F1 confidence intervals we use a cluster bootstrap that resamples at the screenplay level rather than the instance level, accounting for within-screenplay correlation. Each of 10^4 iterations draws 12 screenplays with replacement from the test set's 12 and recomputes the paired macro-F1 difference on all instances from the drawn screenplays; the interval is the 2.5th–97.5th percentile of that distribution. Two implementation details matter for reading the numbers. The macro average is taken over the gold classes of the full test split, held fixed across resamples: recomputing the class set inside each draw would change the denominator with the draw, so the resulting distribution would not belong to a single estimand. And we report percentile intervals rather than BCa, since the acceleration constant would be estimated from 12 clusters.

Clustering effects beyond F1. Screenplay-level clustering affects accuracy and the harm/repair rates as well as macro-F1, so treating it as an F1-only concern would understate uncertainty on the metrics that carry the paper's claims. The 240 test instances come from 12 screenplays, exactly 20 instances each, and instances from one screenplay share a writer, register, and set of characters; correct and incorrect predictions therefore cluster.

We report cluster-robust inference for the confirmatory comparisons rather than for macro-F1 alone. For accuracy and harm/repair we supplement the instance-level McNemar test with the screenplay-level bootstrap described above. Table 15 reports both, so the effect of clustering is visible per metric instead of asserted in general, along with the design effect: the ratio of the cluster-robust variance to the instance-level variance, which quantifies how far the effective sample size falls below 240. The design effect is not uniformly above one, since for L3+NFP versus L3 it is 0.70 on intent and 0.60 on theme, because a paired difference that is consistent across screenplays varies less under a screenplay-level resample than an instance-level calculation assumes. We report the measured values rather than presenting clustering as a one-directional inflation of uncertainty.

Two caveats remain. First, the cluster bootstrap over 12 clusters is itself approximate, since bootstrap coverage degrades when the number of clusters is small; with only 12 screenplays the intervals should be read as indicative rather than exact, and a study designed for cluster-robust inference would want considerably more screenplays in the test split.

Second, the two inferential routes answer different questions and we do not merge them. The Holm-corrected McNemar tests in Table 14 treat the 240 instances as the population of interest and control familywise error across the confirmatory family; the cluster-robust intervals in Table 15 ask whether the same differences would persist over a fresh draw of screenplays, and are uncorrected. A claim is reported as supported only when the Holm-adjusted test rejects *and* the cluster-robust interval excludes zero; where the two disagree we report the disagreement rather than selecting the more favorable one.

Forced-choice instances and the unavailable sensitivity analysis. The annotation prompt requires one of the 8 intent codes and one of the 9 theme codes on every instance. It offers no way to

Table 15: Instance-level versus cluster-robust inference for the confirmatory comparisons. The design effect is the ratio of cluster-robust to instance-level variance; $n_{\text{eff}} = n_{\text{base}}/\text{design effect}$, where n_{base} is the row's own denominator: 240 for accuracy, and the conditioning subset for harm and repair (the instances the comparison system answers correctly and incorrectly, respectively). A design effect below one means the paired difference is more consistent across screenplays than an instance-level calculation assumes. Conclusions are read off the cluster-robust column.

Metric	Comparison	Δ	Instance 95% CI	Cluster 95% CI	Des. eff.	n_{base}	n_{eff}
RA-NarrAgent against the confirmatory family							
Int. acc.	vs. NarrAgent+NFP	$+1.7$	$[-5.1, +8.4]$	$[-4.6, +7.1]$	0.80	240	299
Int. acc.	vs. L3	-2.1	$[-9.1, +4.9]$	$[-10.0, +6.7]$	1.38	240	174
Int. acc.	vs. L3+NFP	-5.0	$[-12.0, +2.0]$	$[-9.6, -0.4]$	0.44	240	547
Int. acc.	vs. NarrAgent+Scalar	$+2.5$	$[-4.2, +9.2]$	$[-3.8, +9.6]$	0.95	240	253
Thm. acc.	vs. NarrAgent+NFP	-5.0	$[-10.1, +0.1]$	$[-10.0, +0.0]$	0.97	240	248
Thm. acc.	vs. L3	-27.5	$[-34.1, -20.9]$	$[-32.5, -22.5]$	0.59	240	409
What the loop and the NFP each cost against L3							
Int. acc.	NarrAgent+NFP vs. L3	-3.8	$[-11.4, +3.9]$	$[-10.0, +2.5]$	0.67	240	358
Thm. acc.	NarrAgent+NFP vs. L3	-22.5	$[-29.8, -15.2]$	$[-29.2, -15.8]$	0.86	240	280
Int. acc.	L3+NFP vs. L3	$+2.9$	$[-5.5, +11.4]$	$[-3.8, +10.0]$	0.70	240	344
Thm. acc.	L3+NFP vs. L3	-28.3	$[-34.6, -22.1]$	$[-32.9, -23.3]$	0.58	240	411
Harm rate	NarrAgent+NFP vs. L3 (thm.)	89.0	—	—	—	82	—
Repair rate	NarrAgent+NFP vs. L3 (thm.)	12.0	—	—	—	158	—

record that none of the categories fits, no free-text alternative, and no mismatch flag, and the two independent annotators of Appendix B were given the same closed choice. Every label in LitSubText is therefore a forced choice, and instances where the taxonomy genuinely fails are not distinguishable from instances where it applies cleanly.

This matters because strict single-label accuracy is the wrong instrument for such instances: the gold label would record the closest available category rather than a category any annotator judged correct, so a system can be penalized for a prediction no worse than the reference. The natural check is to recompute the comparisons with those instances excluded, and under a multi-label rule that accepts either the assigned category or the recorded alternative. **Neither is computable on the current data**, because no instance carries the flag that either treatment would condition on. We report no such analysis rather than estimate the affected set from something else.

What the data does show is where the taxonomy is under strain, which is a weaker but real substitute. The per-class agreement of Table 5 identifies the classes whose boundaries independent annotators do not reproduce, RG at $\kappa = 0.201$ and NS at 0.322 on intent with EM and SE near 0.40 on theme, and the modal-confusion column names what they are confused with. Conclusions that rest on those classes should be read as provisional for that reason. Adding an explicit "no category fits, nearest is X" response to the annotation schema would make the sensitivity analysis available in a future collection round; it is a schema change, not a re-analysis, so it cannot be applied retroactively to these 2,606 instances.

Selection of the 240 test instances. The split is by screenplay, not by instance. We group all 2,606 instances by their source screenplay and partition the 132 screenplays into 108 train / 12 validation / 12 test with seed 42, so that every instance from a given screenplay lands in exactly one split and no writer's style is shared across the train/test boundary. The test split is whatever the 12 test screenplays contain, which is 240 instances; there is no subsampling step and no held-out remainder. The partition was fixed before any system was evaluated.

This design trades statistical power for a stronger independence guarantee. Because instances cluster within screenplays, 240 instances drawn from 12 screenplays carry substantially less information than 240 independent draws, which is why every interval we report on the test set uses the screenplay-level cluster bootstrap described in this appendix rather than an instance-level bootstrap. The class proportions of the test split nonetheless track the corpus despite the split being made at the screenplay level rather than stratified on class (Appendix A).

I The Evidence Grounding Score and Its Limits

EGS is a lexical overlap metric, not an inference-based one. Earlier drafts of this work described it as an NLI pipeline; the implementation is simpler, and the difference matters because a lexical metric and an entailment metric fail in different ways and support different claims.

The source text is the concatenation of the instance's scene description, every dialogue turn, the target utterance, and the retrieved passages the system conditioned on, empty when CGR gated retrieval. Source and explanation are lowercased, tokenized on word boundaries, reduced to sets of content tokens by removing a 60-word stopword list, and compared as sets, so token order, syntax, negation and speaker identity all drop out. The explanation is split into claims by sentence boundary, discarding fragments of ten characters or fewer. A claim counts as grounded when at least half of its content tokens appear anywhere in the source set, EGS is the proportion of grounded claims, and the reported figure is the unweighted mean over instances.

A set-overlap test at a 0.5 threshold detects one failure, an explanation whose vocabulary is largely absent from the evidence, which is the signature of fabricated specifics. It cannot detect a claim that contradicts the evidence, since "Marlowe is not lying to her" and "Marlowe is lying to her" have nearly identical token sets. It cannot separate a defeasible inference from an unsupported assertion, because both are judged only on whether their words appear in the source. And it rewards quotation, so an explanation restating the dialogue verbatim scores perfectly while asserting nothing.

Table 16: EGS under three grounding thresholds, with the mean absolute number of unsupported claims per answer at 0.5. Absolute values necessarily fall as the threshold rises; what matters is that the ordering runs opposite to accuracy, which is why we draw no comparison from this metric.

System	EGS @0.3	EGS @0.5	EGS @0.7	Unsup./answer @0.5
L0 (zero-shot)	0.33	0.09	0.03	1.8
L3	0.21	0.02	0.01	1.5
L3 + NFP	0.17	0.01	0.00	1.1
NarrAgent	0.17	0.02	0.00	1.1
RA-NarrAgent	0.16	0.01	0.00	1.3

The measured values settle the question in the negative and we report them rather than quietly dropping the metric. At the 0.5 threshold EGS is 0.09 for zero-shot and 0.01 to 0.02 for every other configuration, so almost no sentence in a typical explanation has half of its content words appearing anywhere in the scene, the dialogue and the retrieved passages combined. The scores also run opposite to accuracy, because a short explanation reusing the dialogue's own words clears the threshold while a pragmatic reading phrased in fresh vocabulary does not. Both facts follow from the third limitation above, and the ordering is stable across all three thresholds, which is what makes the conclusion a property of the metric rather than of the cutoff. We therefore treat EGS as uninformative for comparing these systems and use it in no claim. Over-interpretation in this domain often consists of elaborate claims built from the passage's own vocabulary, which is exactly where a lexical test is weakest, so the over-interpretation claims rest instead on the critic and control probes of section 5.4–5.4, which vary a candidate against a fixed instance and do not depend on lexical overlap.

J Computational Cost Analysis

J.1 Inference Cost

Table 17 reports the computational cost of each configuration at inference time on the 240-instance test set.

The restraint components were expected to lower cost, and they do not. RA-NarrAgent is the most expensive configuration in the table at 6,490 tokens per instance, 27.1% above NarrAgent and 12.2 times L0 in tokens, or 18.2 times in wall-clock. The CGR gate does shorten the context on the 39 instances where it withholds a passage, and because CGR still scores every passage the saving is in context length rather than retrieval computation, so it is small next to what the loop spends elsewhere.

Table 17: Inference cost per configuration on the 240-instance test set, from the per-instance meter recorded in every results file. Tokens are total controller tokens processed (input plus output) divided by 240, and the critic's own forward passes are counted separately in the last column because they run on a 3B model rather than the 7B controller. Steps and tool calls are per-instance means. Time is wall-clock on a single NVIDIA RTX 5090, summed over the run, so it includes tool-side generation. The **Rel.** column is the *token* ratio against L0, which is the hardware-independent quantity. Wall-clock ratios are larger, 18.2× for RA-NarrAgent against L0, because the loop issues several short generation calls per instance and pays per-call overhead the token count does not capture. The two prompt-only rows take no tool calls by construction, and L3+NFP is listed with the agent entry-point's single forced generation rather than 0 steps because that is how it was run.

Configuration	Steps	Tokens	Tools	Time (min)	Rel.	Critic tok.
L0 (zero-shot)	0	532	0	9.5	1.0×	0
L3 (narrative ctx.)	0	1,157	0	24.8	2.2×	0
L3 + NFP	1	707	0	7.4	1.3×	0
NarrAgent	3.51	5,105	2.51	160.3	9.6×	37.6
NarrAgent-Short	2.88	3,521	1.88	111.1	6.6×	36.0
NarrAgent + NFP	3.92	6,397	2.92	174.7	12.0×	79.5
NarrAgent + Scalar	3.98	6,463	2.98	172.6	12.1×	92.8
RA-NarrAgent	4.02	6,490	3.02	173.9	12.2×	86.8

The IDC concludes early only when the critic signals simplification or sustained maintenance, and SIMPLIFY never fired on this test set, which leaves the mean trajectory at 4.02 steps against 3.51 for the unrestrained loop. The ordering of the four loop rows tracks their step counts almost exactly, which is the expected relationship and the reason the step count is the quantity to control if cost matters.

Two rows are worth separating from that pattern. NarrAgent-Short halves the budget and lands at 3,521 tokens, the cheapest loop configuration, at 11.7% intent macro-F1 against 8.8 to 15.5% for the four full-budget loops. On intent the extra steps therefore buy nothing that a third of the budget does not already reach. On theme the picture is different, since NarrAgent+NFP at 13.0% is well clear of Short at 5.0%, so the budget is not uniformly wasted across the two axes. L3+NFP is cheaper than L3 despite using the same context, at 707 tokens against 1,157, because the null-first instruction produces shorter explanations, and that row is a reminder that a prompt change can move cost in the opposite direction from what adding text to a prompt suggests.

The cost table should be read against the accuracy result rather than as a cost-benefit argument. Because the prompt-only configurations outperform every agent configuration on both label axes (section 5), the cheapest configuration in this table is also the most accurate one. The comparison that matters for a practitioner is L0 against the agent rows, which cost an order of magnitude more and return less. What the table supports is the negative claim that the restraint components add cost without recovering accuracy, which is the same conclusion section 5.2 reaches from the trajectories.

J.2 Training Cost

Training consists of three stages, each with different computational requirements:

Table 18: Training cost breakdown. All training uses a single NVIDIA RTX 5090 (32GB). Batch size is reported as the effective batch after gradient accumulation.

Stage	Base Model	Epochs	Batch Size	Time (h)	GPU-hours
Retriever fine-tuning	`bge-m3` (XLM-R-large)	3	32	1.3	1.3
BiCritic training	Qwen2.5-3B-Instruct	3	16	0.8	0.8
Agent SFT	Qwen2.5-7B-Instruct	2	16	0.2	0.2
Total				2.3	2.3

All three training stages are LoRA adaptations and none dominates the budget. The retriever is the largest single cost at 1.3 GPU-hours over 391 steps per epoch, the BiCritic takes 0.8 hours for 2,273 steps over 12,124 preference pairs enumerated from the 2,126 training instances, and the agent SFT is the cheapest at 0.2 hours for 254 steps over the 2,019 template-synthesized ReAct trajectories (Stage 3, §4.5). The pipeline therefore trains in 2.3 GPU-hours on one consumer 32GB GPU. Training cost is not where this method is expensive; inference is, and Table 17 is the relevant comparison.