

Retrospective Think-Aloud Data as a Window into Performance Trust

A Human-Agent Collaborative Game Study

Adrian Todorov

Supervisor(s): Dr. Myrthe Tielman, Charlotte Ning

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Adrian Todorov
Final project course: CSE3000 Research Project
Thesis committee: Dr. Myrthe Tielman, Bernd Dudzik, Charlotte Ning

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Trust between users and autonomous agents is dynamic, yet most measurement instruments capture only a static end-state. Think-aloud protocol offers a window into the reasoning behind trust judgements, but no prior work has examined how retrospective think-aloud (RTA) data relates to performance trust. This paper presents a user study in which 30 participants interacted with a scripted AI agent in a collaborative box-moving game, verbalised their thoughts while watching a recording of their own gameplay (RTA), and then completed the Multidimensional Measure of Trust (MDMT). A deductive content analysis grounded in the MDMT Reliable and Capable subscales produced a 16-code scheme applied to all 30 transcripts. Negative trust language in RTA tracked the game's scripted trust trajectory, peaking during the most severe failure phase, but did not relate to participants' final MDMT scores - suggesting RTA's diagnostic value lies in the timing of this language across the session, rather than its overall volume. These findings support RTA as a method for capturing within-session trust dynamics that static questionnaires cannot surface.

1 Introduction

Trust is a central factor in human-agent interaction. As autonomous agents become more common in collaborative settings, the ability to detect and respond to miscalibrated trust becomes critical: a user who over-trusts an agent may rely on it beyond its actual capability, while a user who under-trusts it may reject useful assistance [1]. Lee and See (2004) define trust as the attitude that an agent will help achieve an individual's goals in a situation characterised by uncertainty and vulnerability, and distinguish it from reliance: trust is an internal attitude that influences behaviour, but the two do not always align [1].

Trust is not a single construct. Malle and Ullman (2021) identify distinct dimensions including performance trust, related to whether the agent is seen as reliable and capable, and morality trust, related to whether it is seen as honest and fair [2]. These dimensions can develop along different trajectories and respond to different agent behaviours [3]. This paper focuses on performance trust, for two reasons. First, it is the dimension most directly implicated in the over-trust and under-trust costs described above: a user's willingness to rely on an agent plausibly depends heavily on their assessment of its reliability and capability, alongside their judgements of its honesty or fairness. Second, it is more behaviourally groundable: capability and reliability failures – a dropped box, a missed cue, a failed hand-off – can be scripted with an unambiguous link to the intended manipulation, even though the same events could also be read through a moral lens (a missed cue interpreted as a deliberate slight rather than an error). Such evaluations also form relatively early in human-agent interaction [3], making performance trust the dimension most likely to exhibit measurable, single-session dynamics. The MDMT [2] provides a validated instrument for capturing both performance and morality trust, with separate subscales for each. However, like most trust questionnaires, it is administered after the interaction ends: it captures where trust ended up, but not when it shifted, why, or which agent behaviours caused it.

Researchers have explored real-time alternatives, including physiological signals [4] and eye-tracking [5]. These allow continuous monitoring but require specialised hardware and do not distinguish between trust dimensions. In-session self-report prompts offer a lighter-weight approach, capturing trust ratings at key moments without interrupting the task.

Think-aloud (TA) protocol offers a further complementary approach. By asking users to verbalise their thoughts while reviewing a recording of their own interaction – retrospective think-aloud (RTA) – researchers gain access to the reasoning behind user decisions as they reflect on them [6]. Prior work on TA protocols suggests this verbalised content can capture exactly the kind of dynamically changing, evaluative reasoning that a single post-hoc questionnaire cannot [7] [8] – plausibly including the single-session dynamics that the early-forming, measurable nature of performance trust (noted above) would make RTA well placed to surface. We review this evidence, and RTA's known limitations, in detail in Section 2.3.

Despite this potential, no prior work has directly examined how RTA data relates to performance trust specifically, or how well retrospective verbal content corresponds to validated performance trust scores. This paper addresses that gap. Participants in this study interacted with an AI agent in a collaborative game, completed an RTA session, and then rated their trust using the MDMT, administered after the RTA so its items would not prime participants' verbalisations during reflection (full procedure in Section 3).

If RTA can be shown to track performance trust dynamics reliably, it offers a lightweight, hardware-free method for diagnosing not only *whether* trust changed, but *which agent behaviours* drove the change and *how* users reasoned about them – valuable during iterative agent design, where understanding the triggers of trust violations can inform which behaviours to prioritise for correction.

Research Questions

The main research question is: *How does retrospective think-aloud (RTA) data reflect a user's performance trust in an agent?*

This is examined through three sub-questions:

- **SQ1:** What themes or linguistic features appear in RTA data when users express high versus low performance trust?
- **SQ2:** Which interaction events prompt performance trust verbalisations in RTA, and how do these relate to the agent's in-game behaviour?
- **SQ3:** How well do RTA verbal expressions align with MDMT performance subscale scores?

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 describes the method-

ology. Section 4 presents results. Section 5 discusses the findings. Section 6 concludes, followed by a statement on responsible research.

2 Background and Related Work

2.1 Trust as a Multidimensional Construct

Building on the definition of trust introduced in Section 1, a central implication of Lee and See's (2004) framework is calibration: the degree to which a user's trust matches the agent's actual capability [1]. Miscalibrated trust is costly in both directions: over-trust leads users to rely on an agent beyond its actual capability, with potentially serious consequences if it fails, while under-trust leads users to disuse a capable agent, forfeiting its benefits. Calibration is therefore not about maximising trust, but about aligning it with the agent's true reliability.

Trust is not monolithic. Malle and Ullman (2021) decompose it into separate dimensions, operationalised in the Multidimensional Measure of Trust (MDMT), a 16-item validated questionnaire [2]: performance trust (Reliable and Capable subscales, e.g. *reliable*, *predictable*, *capable*, *skilled*) and moral trust (Sincere and Ethical subscales, e.g. *honest*, *ethical*, *transparent*). This separation matters because the two dimensions can move independently [3]: an agent perceived as highly capable but morally suspect, or vice versa, would receive divergent scores despite a similar overall impression of "trustworthiness." For researchers studying how specific behaviours shift trust, this multidimensionality is essential: a single aggregate score would obscure which dimension was affected, and since particular behaviours are expected to map onto particular dimensions, this makes it harder to identify which behaviour caused an observed change.

2.2 Measuring Trust Dynamically

The MDMT, like most validated trust questionnaires, is administered as a single post-hoc measurement, capturing the end-state of trust but no information about how it evolved, which events shifted it, or in which direction. Researchers have explored several approaches to capturing this missing temporal dimension.

Physiological measures, including galvanic skin response and heart-rate variability, have been used to track trust-relevant arousal continuously [4], and eye-tracking to infer moment-to-moment attention and suspicion toward an agent's actions [5]. Both offer fine-grained temporal resolution without self-report, but require specialised hardware and, critically here, do not straightforwardly distinguish performance from moral trust. A spike in arousal at a failure event could plausibly reflect incompetence, dishonesty, or simply surprise, and physiological signals alone cannot easily disambiguate these.

2.3 Think-Aloud Protocols for Performance Trust

Think-aloud protocols offer a different route to the same goal of capturing trust's temporal dynamics: rather than rating trust at intervals, the participant verbalises their ongoing reasoning, and trust-relevant content is identified within that verbalisation after the fact. Alhadreti and Mayhew (2018) compared concurrent think-aloud (CTA), verbalising while performing a task, and retrospective think-aloud (RTA), verbalising while reviewing a recording of a completed task [6]. CTA detected more minor usability problems, while RTA was more susceptible to memory decay for such transient details; critically, neither condition showed reactivity effects on task performance.

This finding, however, comes from a sequential, screen-based usability task and may not generalise to denser real-time demands. The present game requires continuous, time-pressured coordination, plausibly leaving little spare attentional capacity for CTA's simultaneous verbalisation. RTA sidesteps this by design, since verbalisation occurs only once attentional demands have passed – making it the more practical choice here even without direct evidence CTA would disrupt performance specifically.

The susceptibility of RTA to memory decay for minor events does not imply equal susceptibility for all content types. Pathan et al. (2025) found that RTA consistently surfaces reflective and evaluative content, including causal attributions of the general kind central to judging an agent's reliability and capability [8]. Cutler et al. (2025) further show that think-aloud data captures a dynamically changing state of mind that a single post-hoc questionnaire cannot represent [7]; applied to trust, this could include moments where a participant's trust is extended and subsequently retracted within a single session. Taken together, while RTA may lose some low-salience detail, the high-salience evaluative content most relevant to performance trust is well preserved, making RTA a promising lens for the present study.

3 Methodology

3.1 Game and Stimuli

To investigate how RTA reflects performance trust, we designed a controlled human-agent collaborative task in which the agent's reliability and capability could be manipulated on a known schedule, so that trust-relevant verbalisations could be tied to specific, scripted events. The study uses a modified version of the Moving Out collaborative game, originally implemented in MATRX (a Python framework for human-agent team research) and validated in prior trust research by Centeio Jorge et al. (2023) [9], who showed that scripted automation failures in this environment produce significant, measurable changes in subjective trust. A human participant and an AI agent collaborate to move boxes from a shared workspace into a marked dropzone under a time limit (Figure 1). Three box types require different handling: light boxes can be carried alone by either participant, medium boxes alone at a speed penalty or jointly at full speed, and heavy boxes only jointly. The team's score increases by 10 points per box, with a five-point bonus for boxes delivered in the order indicated by the game's ordering cues.

The session is divided into four phases designed to produce a controlled trust trajectory. Phases 1 and 3 are trust-building phases in which the agent performs without any scripted failures, behaving reliably and capably. Phase 2 introduces a set of mild scripted failures intended to produce an initial, moderate drop in trust. Phase 4 introduces more serious scripted

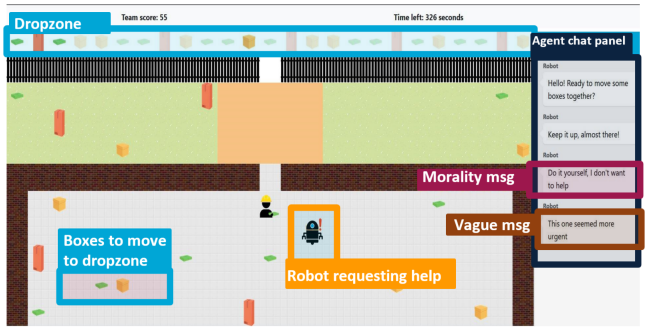


Figure 1: Screenshot of the modified Moving Out environment. The participant (left) and the AI agent (right) collaborate to move light, medium, and heavy boxes from the shared workspace into the marked dropzone.

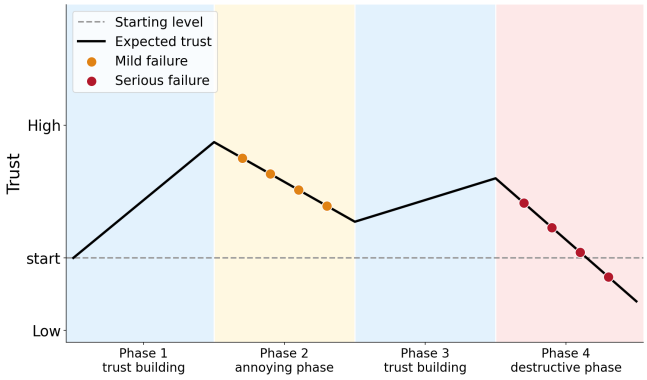


Figure 2: Expected performance trust trajectory across the four game phases: a build-up during Phase 1, a moderate drop following the mild failures of Phase 2, partial recovery during Phase 3, and a more severe drop following the serious failures of Phase 4.

failures, intended to drive trust below its starting level. Figure 2 illustrates this expected trajectory.

Across Phases 2 and 4, the agent exhibits eight scripted failure events, drawn from six failure types. Each is paired with a specific type of in-game chat message: a *performance-framed* message explicitly attributing the failure to a capability or system issue, a *morality-framed* message framing it as a deliberate choice, or a *vague* message attributing no cause at all. Table 1 lists the six failure types, the phase(s) in which they occur, their framing, and an example message.

These manipulations serve two purposes: the alternation between failure-free (1, 3) and failure phases (2, 4) creates the controlled trajectory needed to test SQ2 – a build-up, moderate drop, partial recovery, and more severe drop – while the three message framings let the study examine whether the attribution type offered influences the trust-relevant language produced.

Prior work has shown that agent messages following failures influence subsequent trust dynamics: the framing of a failure – attributed to incompetence, deliberate choice, or left ambiguous – affects how users respond and recover their trust [10]. This motivates the typed, framed agent messages above, examined directly via H4 (Section 3.2).

Table 1: Scripted failure types, the phase(s) in which they occur, their message framing, and an example agent message.

Failure type	Phase	Framing	Example message
Slow movement	2	Performance	"My systems are running slow"
Ignore help	2, 4	Morality	"Do it yourself"
Wrong order	2	Vague	"This one seemed more urgent"
Fake help req.	2	Vague	"Could use some help here"
Break box	4 ($\times$2)	Vague	"Oops!"
Wrong dropzone	4	Performance	"I miscalculated the target"

3.2 Hypotheses

Based on the background above, four hypotheses are tested.

H1 (SQ1): RTA transcripts will contain identifiable linguistic features reflecting performance trust, including evaluative statements about the agent's reliability and capability, causal attributions for its failures, and explicit moments where the participant updates their trust.

H2 (SQ2): Performance trust verbalisations will cluster around scripted failure events, with the most severe failures in Phase 4 producing the highest density of negative performance trust language of any phase.

H3 (SQ3): The frequency of negative performance trust statements in RTA will correlate with MDMT Reliable and Capable subscale scores.

H4 (exploratory): Vague agent messages will produce more extensive performance trust verbalisations than clearly framed messages, as the ambiguous framing requires participants to form their own attribution.

3.3 Participants

30 participants completed the RTA condition analysed in this paper. Data collection was split evenly between two researchers following the same protocol, each independently running 15 sessions. Participants were recruited from academic and non-academic backgrounds rather than exclusively technical disciplines, to avoid over-representing individuals with prior AI exposure, which could bias trust judgements toward more technically grounded attributions. The sample comprised 19 male and 11 female participants, predominantly aged 20–25.

Each session produced five data artefacts: a silent screen recording of the gameplay, an audio and screen recording of the RTA session, four in-session trust pop-up ratings, a completed MDMT questionnaire, and a game log recording all in-game events and their timestamps. Participants are identified throughout by a numeric ID only; no names are stored alongside any recording, transcript, or questionnaire response.

The study followed TU Delft's Human Research Ethics guidelines. All participants gave written informed consent and could withdraw at any point without reason or consequence. To avoid priming participants' behaviour or verbalisations, the scripted failures' nature and timing were not dis-

closed beforehand. The task carries minimal risk, comparable to a casual video game; the only anticipated discomfort – mild, intentional frustration – is explained during debriefing.

3.4 Procedure

Each session followed a fixed procedure, in person or remotely: participants read a briefing, signed informed consent, received a short tutorial on the game's controls and scoring, then completed an initial trust pop-up (T_0) as baseline.

The main session consisted of the four phases described in Section 3.1. Immediately after each of Phases 1–3, participants completed a further trust pop-up (T_1–T_3 respectively). No pop-up was administered after Phase 4, as it led directly into the RTA session.

Following Phase 4, the participant watched a silent recording of their own gameplay with the researcher present, verbalising aloud anything recalled, pausing the video at will; the researcher occasionally prompted ("What were you thinking here?") when needed. This was audio-recorded for transcription.

Finally, participants completed the MDMT questionnaire, administered after rather than before or during the RTA so its own items (e.g., *reliable*, *capable*, *honest*) would not prime the vocabulary participants used during verbalisation.

3.5 Measurements

Participants rated a Morality and a Performance item on a 0–7 scale at each pop-up point described above; this paper analyses the Performance item (Section 4.2), while the Morality item is analysed separately in parallel work. RTA audio was transcribed and segmented into time-aligned utterances, the primary data source for the coding in Section 3.6. After the RTA, participants completed the full 16-item MDMT (Reliable/Capable and Sincere/Ethical subscales, 0–7); this paper uses the Reliable and Capable subscales and their mean (performance composite) as dependent measures for SQ3. Each session also produced a timestamped game log of scripted failure onsets, used to align RTA utterances with the failure being discussed (Section 4.4).

3.6 Analysis

The coding scheme was developed using a hybrid deductive-inductive approach. Initial categories were derived deductively from the MDMT Reliable and Capable subscale items (e.g., *reliable*, *predictable*, *capable*, *skilled*). These were refined and extended inductively through an initial pass over a subset of transcripts, adding categories for content the deductive scheme missed – calibration statements, reactions to events, attributions of intent, and behavioural adaptations. The resulting 16-code scheme (full codebook with examples in Table 2, Appendix) is organised into five categories – Performance, Morality, Calibration, Event Reaction, and Behaviour – together capturing the linguistic features anticipated by H1: evaluative vocabulary, causal attribution language, explicit trust-update statements, and behavioural descriptions of adaptation.

For inter-coder reliability, a second coder independently applied the codebook to six transcripts (20% of the sample); raw agreement, the Brennan-Prediger coefficient, Cohen's κ, and conditional Jaccard overlap were computed (Section 4.1).

For SQ2, code frequencies were compared across the four game phases using paired t-tests, separately for positive and negative codes (Section 4.3), and across the eight scripted failure events (Section 4.4). For H4, performance-trust code frequencies were additionally compared across the three message-framing categories in Table 1 (Section 4.5). For SQ3, per-participant code frequencies were correlated with MDMT Reliable and Capable subscale scores using Spearman's rank correlation, chosen given the sample size ($N = 30$) (Section 4.6). The qualitative analysis (SQ1, Section 4.7) carries the primary explanatory weight throughout; the quantitative analyses serve as a convergence check on the patterns it identifies.

4 Results

4.1 Inter-coder Reliability

All 30 transcripts were coded by the author as the primary coder; a second coder independently applied the codebook to six of these (Participants 301, 304, 314, 401, 411, 415) for inter-coder reliability. Agreement was assessed per line-by-code decision, since each line could receive multiple codes: for every line and code, each coder either applied it or did not (5,696 binary cells). Raw observed agreement was 0.96 and the Brennan-Prediger coefficient [11] was 0.92, while Cohen's κ [12] was 0.53. The gap between these reflects a difference in coding density rather than disagreement over meaning: the primary coder applied 331 codes on these six transcripts against the second coder's 187. Because κ estimates chance agreement from marginal frequencies, this imbalance deflates κ despite high raw agreement; since this denser style was applied across all 30 transcripts, it has implications for the dataset as a whole, addressed in Section 5.2. Conditional on both coders tagging the same line, mean Jaccard overlap was 0.71, indicating the coders largely agreed on which code applied, with disagreement mainly about whether a line warranted a code at all.

4.2 In-Session Trust Trajectories

Figure 3 shows the mean performance trust pop-up score at each rating point across all 30 participants, with individual trajectories in the background. Mean scores were $T_0 = 4.63$ ($SD = 1.22$), $T_1 = 5.50$ ($SD = 1.01$), $T_2 = 4.20$ ($SD = 1.86$), and $T_3 = 4.73$ ($SD = 1.28$), all on a 0–7 scale. Paired t-tests confirmed that each transition was statistically significant: $T_0 \rightarrow T_1$ (Phase 1 build): $t(29) = -4.18$, $p < .001$; $T_1 \rightarrow T_2$ (Phase 2 failure): $t(29) = 4.28$, $p < .001$; $T_2 \rightarrow T_3$ (Phase 3 recovery): $t(29) = -2.64$, $p = .013$. The direction and significance of all three transitions match the expected trust trajectory (Figure 2), confirming the manipulation produced the intended dynamics and establishing the variance needed for the SQ3 correlation.

No pop-up was administered after Phase 4, so Figure 3 additionally plots the post-session MDMT performance composite as a fifth point, connected to T_3 by a dashed segment to mark the change of instrument. This composite ($M = 2.93$, $SD = 1.04$) sits notably below T_3, a divergence we return

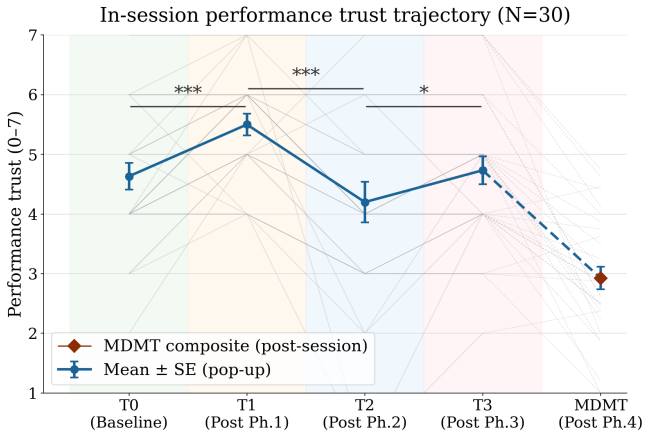


Figure 3: Mean in-session performance trust ratings (T_0–T_3, solid) with the post-session MDMT performance composite added as a fifth point (dashed segment, diamond marker), individual participant trajectories (grey), and mean ± SE (blue). Significant transitions are marked for the matched-instrument pop-up comparisons; *** $p < .001$, * $p < .05$.

to in Section 5.1. This trajectory is revisited in Section 5.1 alongside the phase-based RTA results below.

4.3 Phase-Based Code Dynamics (SQ2)

Figure 4 shows mean positive and negative code frequency per participant across the four game phases. Negative codes tracked the expected trust trajectory closely: mean counts were 0.57 ($SD = 0.86$) in Phase 1, 6.60 ($SD = 3.23$) in Phase 2, 2.40 ($SD = 2.13$) in Phase 3, and 10.80 ($SD = 4.37$) in Phase 4. All phase transitions were significant: Ph1→Ph2: $t(29) = -10.54$, $p < .001$; Ph2→Ph3: $t(29) = 5.75$, $p < .001$; Ph3→Ph4: $t(29) = -11.09$, $p < .001$. The effect from Phase 1 to Phase 4 was the largest observed: $t(29) = -12.67$, $p < .001$.

Positive codes showed a different pattern. Counts were 3.57 ($SD = 2.71$), 4.20 ($SD = 2.62$), 2.10 ($SD = 1.81$), and 2.73 ($SD = 2.27$) across Phases 1–4. The only significant transition was a decrease from Phase 2 to Phase 3: $t(29) = 4.66$, $p < .001$. Phase 1→Phase 2 and Phase 3→Phase 4 transitions were not significant ($p > .10$); this asymmetry between the consistently significant negative-code transitions and the largely flat positive-code counts is discussed in Section 5.1.

These results support **H2**, with RTA language mirroring the trust trajectory captured by pop-up ratings; reviewed in detail, alongside the trajectory data above, in Section 5.1.

4.4 Event-Level Code Distribution (SQ2)

Figure 5 shows the total code count per scripted failure event across all participants. Phase 4 events (BREAK_BOX1 and BREAK_BOX2) generated substantially more negative codes than their Phase 2 counterparts, consistent with the design intent that Phase 4 failures are more severe. Within Phase 2, FAKE_HELP_REQUEST produced the most negative coding activity; within Phase 4, both BREAK_BOX events generated high densities of both negative and positive codes, the

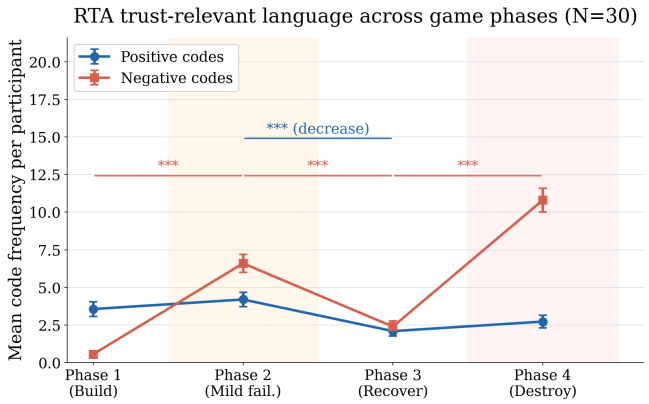


Figure 4: Mean positive (blue) and negative (red) code frequency per participant across the four game phases (mean ± SE). All negative-code transitions are significant at $p < .001$; positive codes show a significant decrease only at the Phase 2→3 transition.

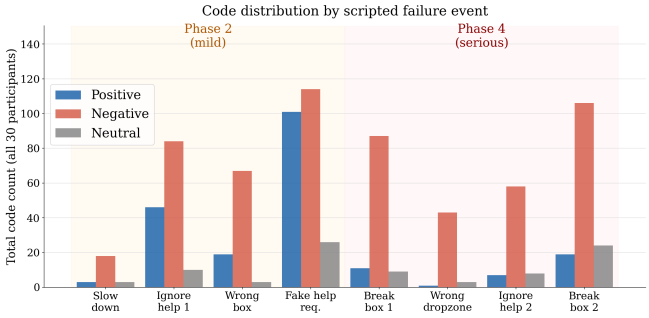


Figure 5: Total positive, negative, and neutral code counts per scripted failure event (all 30 participants). Phase 2 events (left) and Phase 4 events (right) are shaded separately.

latter reflecting surprise or attempts to compensate. The two IGNORE_HELP events generated predominantly negative intent codes (I_N), with participants frequently attributing the agent's non-response to deliberate indifference rather than malfunction, even though the failure message for this event was morality-framed (Table 1). This last observation is examined more systematically in Section 4.5.

4.5 Message Framing and Performance-Trust Language (H4)

H4 (exploratory) predicted that vague failure messages, which leave the cause of a failure unattributed, would elicit more performance-trust verbalisation than messages that explicitly frame the failure as a performance issue (e.g., "My systems are running slow") or a morality issue (e.g., "Do it yourself"). Figure 6 compares the mean number of performance-trust codes (REL_P, REL_N, CAP_P, CAP_N combined) produced per event, grouped by the message-framing category of that event (vague: WRONG_BOX, ASK_HELP_LIGHTBOX, BREAK_BOX1, BREAK_BOX2; performance-framed: SLOW_DOWN, WRONG_DROPZONE; morality-framed: IGNORE_HELP1, IGNORE_HELP2; cf. Table 1).

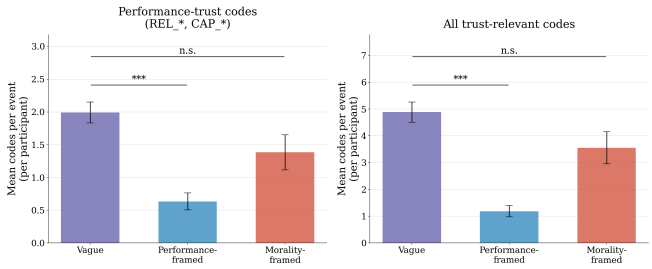


Figure 6: Mean codes per event, grouped by message-framing category: vague, performance-framed, and morality-framed. Left: performance-trust codes (REL_*, CAP_*) only. Right: all trust-relevant codes. Error bars show ± SE; brackets show paired t-test results vs. the vague condition.

Vague events produced significantly more performance-trust codes per event ($M = 1.99$, $SD = 0.88$) than performance-framed events ($M = 0.63$, $SD = 0.71$; $t(29) = 6.60$, $p < .001$). The comparison with morality-framed events ($M = 1.38$, $SD = 1.47$) was in the predicted direction but did not reach significance ($t(29) = 1.99$, $p = .057$). The same pattern held for total code counts of any type: vague events produced more codes overall ($M = 4.88$) than performance-framed ($M = 1.18$; $t(29) = 8.27$, $p < .001$) or morality-framed events ($M = 3.55$; $t(29) = 1.85$, $p = .075$).

These results offer partial support for H4 and are interpreted in Section 5.1.

4.6 Correlation with MDMT Scores (SQ3)

Spearman correlations between per-participant RTA code frequencies and MDMT performance subscale scores were uniformly non-significant. The primary analysis – negative performance codes (REL_N + CAP_N) against the Reliable subscale – yielded $\rho = 0.050$, $p = .792$. Against the Capable subscale: $\rho = -0.064$, $p = .739$. Total negative codes against the performance composite: $\rho = -0.047$, $p = .804$. Total positive codes against the performance composite: $\rho = 0.197$, $p = .296$ (all $n = 30$).

These results do not support **H3**. The frequency of negative RTA statements across the full session does not predict where a participant ends up on the MDMT. This null finding is addressed in Section 5.1.

4.7 Qualitative Patterns in RTA Language (SQ1)

Inspection of coded transcripts across all 30 participants supports **H1**: RTA data consistently contains identifiable linguistic features reflecting performance trust. Three patterns are noted here; these are reviewed against H1 in Section 5.1.

Performance language dominated. REL_N (181 instances) and CAP_N (119) were the two most frequent codes across the corpus, substantially exceeding their positive counterparts REL_P (105) and CAP_P (43). Even following morality-framed failure messages ("Do it yourself"), participants predominantly framed failures in performance terms, attributing the agent's behaviour to unreliability or incompetence rather than intent.

Calibration events are identifiable. Explicit trust update statements (CAL_P, CAL_N, CAL_ST; combined 216 instances) were present in nearly every transcript and tended to cluster around quiz pop-up moments, where the video retrieval cue prompted participants to explain their ratings aloud. This produced explicit trust reasoning that is not captured in the rating alone.

Behavioural adaptation tracks trust. Working Alone (WA; 45 instances) and Changing Strategy (CS; 40 instances) codes appeared predominantly in Phases 2 and 4, indicating that participants verbalised autonomous workarounds in direct response to agent failures. This behavioural language provides a complementary window into trust beyond evaluative statements: not only did participants say the agent was unreliable, they described compensating for it.

5 Discussion

5.1 Interpretation of Findings

The four hypotheses received differing levels of support, ranging from strong support (H1, H2) to a clear null result (H3) to partial, exploratory support (H4). Each is reviewed in turn below, followed by a discussion of how participants conceptualised the agent across the performance and morality dimensions more broadly.

H1: Linguistic Features of Performance Trust

H1 predicted that RTA transcripts would contain identifiable linguistic features reflecting performance trust: evaluative statements about reliability and capability, causal attributions of failure, and explicit trust-update moments. This is clearly supported (Section 4.7). Performance-category codes were the most frequent category overall: REL_N and CAP_N alone (181 + 119 = 300 instances) outnumbered all four morality codes combined (128 instances, discussed further below), and performance codes as a whole substantially exceeded intent-attribution codes throughout the corpus. Because all 30 transcripts were coded by the primary coder, whose coding density differed from the second coder's (Section 4.1), these comparisons should be read as relative densities under one consistent coding style rather than precise absolute counts – though the gap is large enough that coding density alone is unlikely to explain it. Intent-attribution codes (I_P, I_N) captured causal reasoning about the agent's behaviour, and calibration codes (CAL_P, CAL_N, CAL_ST) captured 216 explicit trust-update statements across the 30 transcripts. The coding scheme – developed deductively from the MDMT subscales and refined inductively – applied across all 30 transcripts without substantial post-hoc categories, indicating the linguistic features anticipated by H1 are stable and recognisable in this context. This echoes Malle and Ullman's argument that performance and morality trust are separable, recognisable constructs, rather than a single undifferentiated impression [2].

H2: Phase-Based Clustering and the Trust Trajectory

H2 predicted that performance trust verbalisations would cluster around scripted failure events, with Phase 4 producing the highest density of negative performance trust language. This is strongly supported (Section 4.3): negative code frequency rose from 0.57 in Phase 1 to 6.60 in Phase 2, fell to 2.40 in Phase 3, and rose again to 10.80 in Phase 4 – the

highest of any phase, and roughly $1.6\times$ the Phase 2 level. All transitions were significant at $p < .001$.

This trajectory closely matches that anticipated in Figure 2: a build phase, a moderate failure-driven drop, a partial recovery, and a more severe final drop. The in-session pop-up ratings (Section 4.2) follow the same pattern for T_0–T_3, and matter beyond simply confirming the manipulation worked: without a numeric measure at the same granularity as the RTA phases, the divergence described next would not have been detectable, since a purely qualitative reading could have suggested recovery without any way to quantify how complete it was. The pop-ups are, then, the precise anchor against which RTA's verbal record can be checked for agreement or divergence.

That divergence is itself notable. The pop-up rating recovers fully by T_3 ($M = 4.73$, vs. $T_0 = 4.63$), yet RTA negative-code density in Phase 3 remains over four times the Phase 1 level (2.40 vs. 0.57), and the post-session MDMT composite ($M = 2.93$) sits well below T_3 despite both nominally measuring the same construct moments apart. Together, these suggest explicit, in-the-moment ratings recover faster than either the underlying verbal processing of a violation or the post-hoc judgement that follows – echoing Cutler et al.'s observation that think-aloud data can surface a more conflicted state of mind than a single rating implies [7], and consistent with trust-repair accounts that treat recovery as gradual rather than a discrete reset [10]. This divergence is itself evidence of RTA's diagnostic value: a single recovered rating at T_3 would not have revealed it, whereas the phase-resolved RTA data does.

H3: Correlation with MDMT Scores

H3 predicted that the frequency of negative performance trust statements in RTA would correlate with MDMT Reliable and Capable subscale scores. This was not supported: all Spearman correlations tested were non-significant, with ρ values close to zero in both directions (Section 4.6).

This null result should be read alongside the strong support for H2: RTA content clearly tracks trust dynamics within a session, but aggregating it into a single per-participant count does not predict the end-state MDMT score. Two factors likely contribute. First, the MDMT and RTA frequency may capture related but distinct constructs: the MDMT is a deliberative, post-hoc summary judgement, while RTA frequency sums a temporally structured sequence of reactions whose order and recency may matter more than its total – collapsing precisely the temporal structure that Section 5.1 identifies as where RTA's signal lies. Second, individual differences in verbosity mean a raw frequency count conflates verbal output style with trust evaluation, further suppressing the correlation.

This second factor is, in principle, controllable. Rerunning the correlation using each participant's negative-code count as a *proportion* of their total code output, rather than as a raw count, partially addresses verbosity differences. This proportion-based measure showed a numerically stronger relationship with the performance composite ($\rho = -0.262$, $p = .161$) than the raw count did ($\rho = -0.047$, $p = .804$), though it still did not reach significance at $n = 30$. This suggests verbosity is a genuine, if partial, suppressor of the relationship – though any single-number summary, proportional or not, still collapses the temporal structure that Section 5.1 identifies as the location of RTA's diagnostic value.

The null result does not undermine the paper's central argument; if anything it sharpens it: RTA's value lies in revealing the *trajectory* of trust, not a scalar summary that duplicates the MDMT. This is in line with calibration accounts of trust as a continuously updated state rather than a fixed trait [1]: a method that aggregates away the update process is poorly suited to testing claims about how that process unfolds. Future work exploring time-weighted or phase-specific RTA metrics, rather than raw or proportional totals, may recover a relationship with MDMT scores (Section 5.3).

H4: Message Framing

H4, posed as exploratory, predicted that vague agent messages would produce more extensive performance-trust verbalisation than clearly framed (performance- or morality-framed) messages, since ambiguous framing requires participants to construct their own attribution. This received partial support (Section 4.5): vague events produced significantly more performance-trust codes per event than performance-framed events ($p < .001$), and the comparison with morality-framed events trended in the same direction without reaching significance ($p = .057$).

The more striking, and at first glance counterintuitive, pattern was that performance-framed messages – which explicitly state a capability-related cause ("my systems are running slow") – produced the *least* performance-trust language of the three categories, despite being the framing most semantically aligned with the performance-trust codes themselves. One plausible explanation is that spontaneous causal reasoning is triggered by surprising or unexplained outcomes, and once an external cause is supplied, little inferential work remains. Vague and morality-framed messages leave that work to the participant for different reasons: vague messages offer no cause at all, while a moral attribution ("do it yourself") is a less complete causal explanation than a stated system fault. This suggests an explicit agent-provided attribution can preempt the participant's own evaluative reasoning rather than amplify it: when the cause is already stated, there is less to verbalise. This qualifies H4 – ambiguity does appear to drive more elaboration, but not because participants "fill in" a performance-relevant explanation that a clear performance-framed message would otherwise have supplied; if anything, the clear message substitutes for that elaboration entirely.

This raises a further question the current data cannot answer: whether reduced verbalisation also means reduced *impact* on trust, or only a reduced tendency to talk about it. Because the pop-ups are collected once per phase rather than once per event, this cannot be checked directly here; together with the caveat that the vague category includes the two visually salient BREAK_BOX events (Section 4.5), both issues point to event-level trust probes as a natural target for future work (Section 5.3).

Performance vs. Morality Language

Independently of message framing, performance language dominated morality language throughout the corpus: REL_N

alone (181 instances) outnumbered all morality codes combined (I_P + I_N + OMS + M_NA = 128). As with the H1 comparison above (Section 5.1), this gap is large enough that it is unlikely to be solely an artefact of the primary coder's denser coding style relative to the second coder's (Section 4.1); even a substantial correction would leave performance codes well ahead. Several participants explicitly declined to attribute intentions or emotions to the agent (M_NA), while others anthropomorphised extensively (OMS); this bimodal pattern suggests individuals vary in how they conceptualise AI agents, which may shape how violations are processed and verbalised. Combined with the H4 finding that morality-framed messages ("Do it yourself") still elicited substantial performance-trust language ($M = 1.38$ codes per event, second only to vague events), this supports the conclusion that participants here predominantly process agent failures – regardless of framing – through a performance lens. This is consistent with the view that morality-based trust may not transfer straightforwardly from human-human to human-automation contexts [9]: even when an agent's behaviour is explicitly narrated in moral terms by the agent itself, participants in this task-focused setting still default back to a performance-based reading. For designers, this suggests that in single-session, task-focused interactions, monitoring and repairing performance trust may be the higher-leverage intervention, even when a failure could reasonably be framed in moral terms.

5.2 Limitations

The findings above should be interpreted alongside several limitations of the current study, spanning sample size, generalisability, measurement, and the coding process itself.

Small sample. $N = 30$ limits statistical power, particularly for the Spearman correlations; the qualitative findings (Section 5.1) carry the primary explanatory weight for SQ1.

Single game context and chat attention. All data come from one collaborative game with one agent and one failure script; generalisability to other tasks, agents, or failure types is untested. Within this design, the limited type-matching effect of message framing (Section 5.1) may also partly reflect inconsistent attention to the agent's chat panel during gameplay, reducing experimental control over the manipulation.

Memory decay. RTA remains susceptible to memory decay for low-salience events [6], though performance trust judgements' persistent, evaluative nature appears to mitigate this for the high-salience failures analysed here [8]; minor moments may still be under-represented.

Coding density asymmetry. The inter-coder reliability assessment (Section 4.1) revealed that the primary coder, who coded all 30 transcripts, applied substantially more codes than the second coder, despite high raw agreement and Jaccard overlap on shared lines. This suggests the codebook's boundary between "codeable" and "non-codeable" content requires further specification before the scheme could be applied reliably by independent coders without additional training.

5.3 Future Work

Temporal and verbosity-controlled RTA metrics. Future work should explore time-weighted or phase-specific RTA metrics – e.g., the ratio of Phase 4 to Phase 1 negative-code frequency, or a recency-weighted sum – as predictors of MDMT scores, building on the partial improvement seen from verbosity-controlled proportions (Section 5.1).

Event-level granularity. A follow-up design holding event type constant while varying message framing would isolate the H4 framing effect (Section 5.1) from event saliency. Pairing this with event-level trust probes and eye-tracking would further test whether reduced verbalisation after performance-framed messages reflects a smaller real shift in trust or only inattention to the chat panel.

Generalisation. This study used a single session, one game, and one agent. Longitudinal designs would test whether the performance-morality distinction strengthens with a longer agent history; applying the scheme to other agent types or tasks would test its reusability; and integrating lightweight RTA-style debriefs into industrial design pipelines – requiring automation of the largely manual coding shown here – is a natural next step for translating this method beyond the lab.

6 Conclusions

This paper investigated how retrospective think-aloud (RTA) data reflects a user's performance trust in an agent, using a modified collaborative game with scripted failures as the experimental platform. Across $N = 30$ RTA sessions, the four hypotheses introduced in Section 3.2 received differing levels of support. **H1** was supported: RTA transcripts consistently contained the anticipated linguistic features – evaluative statements, causal attributions, and explicit calibration moments. **H2** was strongly supported: negative performance-trust language tracked the scripted trust trajectory closely, peaking sharply in the most severe failure phase. **H3** was not supported: aggregate RTA code frequency did not correlate with end-state MDMT scores, indicating that RTA's diagnostic value lies in its temporal structure rather than its total volume. **H4**, exploratory, received partial support: vague failure messages elicited significantly more performance-trust language than performance-framed messages, counterintuitively the opposite of a simple type-matching account. The contributions of this work are threefold. First, a coding scheme for performance-trust markers in RTA, grounded in the MDMT Reliable and Capable subscales and applied consistently across 30 transcripts. Second, an empirical demonstration that RTA language tracks within-session performance-trust dynamics with high sensitivity, including a divergence between RTA and pop-up ratings during the recovery phase, and a further divergence between the final pop-up rating and the post-session MDMT score (Section 5.1), that together highlight RTA's distinct diagnostic value beyond what a single end-of-phase rating can capture. Third, a first empirical characterisation of how agent message framing relates to the volume and content of performance-trust verbalisation, including the counterintuitive finding that explicit performance-framed attributions appear to substitute for, rather than prompt, participants' own evaluative reasoning.

Responsible Research

Ethical considerations regarding participant consent, safety, and debriefing are described in Section 3.3. All data were stored either locally or on TU Delft OneDrive, with access restricted to the author and research collaborators. Audio recordings were transcribed using WhisperX, a local transcription tool, and qualitative coding was carried out in Taguette, a locally-installed annotation tool; in both cases, data were never sent to external servers. The only personally identifiable data collected were the audio recordings, used solely for transcription. Transcripts and anonymised questionnaire responses are retained for replication and peer review.

Use of AI. Generative AI (Claude, Anthropic) was used at several stages of this project. During the design phase, it was used to brainstorm and sanity-check aspects of the experimental design, including the failure-event schedule and message framing conditions. During data preparation, it assisted in identifying and correcting a data-entry offset affecting a subset of pop-up and MDMT scores. During qualitative coding, AI was used to help develop and refine the codebook and to propose codes for individual transcript segments; all AI-proposed codes were reviewed, corrected, and approved by the author, who remained the primary coder of record for all 30 transcripts. During analysis, it assisted with writing and debugging the Python code used for statistical analysis and figure generation. During writing, AI was used to help rewrite and polish drafts of several sections for clarity and academic tone, and to flag weakly-supported claims requiring citation or rephrasing. All substantive analytical decisions, interpretations, and conclusions are the author's own.

A Codebook

References

[1] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004.

[2] B. F. Malle and D. Ullman, "A multidimensional conception and measure of human–robot trust," in *Trust in Human–Robot Interaction*, C. S. Nam and J. B. Lyons, Eds. Elsevier Academic Press, 2021, pp. 3–25.

[3] C.-W. Ning, C. Centeio Jorge, M. L. Tielman, and M. A. Neerincx, ""What's on your mind?": understanding the development of multidimensional trust in social robots," in *Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction*, 2026, pp. 924–933.

[4] J. Ayoub, L. Avetisian, X. J. Yang, and F. Zhou, "Real-time trust prediction in conditionally automated driving using physiological measures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 641–14 654, 2023.

[5] Y. Zhang, A. Yadav, S. K. Hopko, and R. K. Mehta, "In gaze we trust: Comparing eye tracking, self-report, and physiological indicators of dynamic trust during HRI," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 2024, pp. 1188–1193.

[6] O. Alhadreti and P. Mayhew, "Rethinking thinking aloud: A comparison of three think-aloud protocols," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 2018, pp. 1–12.

[7] Z. Cutler, L. Harrison, C. Nobre, and A. Lex, "Crowdsourced think-aloud studies," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, 2025, pp. 1–23.

[8] R. Pathan, S. Murthy, and R. Rajendran, "Identifying cognitive and metacognitive processes using retrospective and concurrent think-aloud protocols," in *International Conference on Computers in Education*, 2025.

[9] C. Centeio Jorge, N. H. Bouman, C. M. Jonker, and M. L. Tielman, "Exploring the effect of automation failure on the human's trustworthiness in human-agent teamwork," *Frontiers in Robotics and AI*, vol. 10, p. 1143723, 2023.

[10] C. Esterwood and L. P. Robert, "Having the right attitude: How attitude impacts trust repair in human–robot interaction," in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 2022, pp. 332–340.

[11] R. L. Brennan and D. J. Prediger, "Coefficient kappa: Some uses, misuses, and alternatives," *Educational and Psychological Measurement*, vol. 41, no. 3, pp. 687–699, 1981.

[12] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

Table 2: Full codebook used for RTA transcript coding, with one illustrative example per code drawn from the coded transcripts.

Category	Code	Name	Description	Example
Performance	CAP_P	Capability – Positive	Participant says the agent is skilled, competent, or makes good decisions	"It's making good decisions."
Performance	CAP_N	Capability – Negative	Participant criticises agent errors, poor decisions, or incompetence	"Actually, the robot dropped it."
Performance	REL_P	Reliability – Positive	Agent behaved consistently, predictably, or as expected	"For now, the robot is pretty reliable."
Performance	REL_N	Reliability – Negative	Agent behaviour was surprising, inconsistent, or unpredictable	"I didn't expect that."
Morality	I_P	Intent – Positive	Participant attributes good will or cooperative intent (e.g. "wanted to", "was trying to help")	"I see that it is also wanting to help me."
Morality	I_N	Intent – Negative	Participant attributes bad intent or deliberate indifference (e.g. "decided to", "didn't want to", "chose to ignore")	"It just didn't want to help."
Morality	M_NA	Morality Not Applicable	Participant treats the agent as purely mechanical, declining to attribute feelings or intentions	"Not that it's making those mistakes willingly, just it doesn't have it in its parameters or something."
Morality	OMS	Other Morality / Humanising	Attributes human-like qualities beyond intent – emotions, personality, learning	"So I thought this robot was quite intelligent and kind of foreseeing."
Calibration	CAL_P	Trust Increase	Participant explicitly describes their own trust as having increased	"So I give the robot high grades."
Calibration	CAL_N	Trust Decrease	Participant explicitly describes their own trust as having decreased or being lost	"I kind of lost trust in him."
Calibration	CAL_ST	Trust Stable	Participant explicitly describes their trust as unchanged or at a baseline level	"I wanted to be neutral."
Event Reaction	ER_P	Positive Reaction	Positive emotional response – relief, happiness, pleasant surprise	"Yeah, I like this."
Event Reaction	ER_N	Negative Reaction	Negative emotional response – frustration, annoyance, disappointment, sadness	"It made me annoyed."
Behaviour	WA	Working Alone	Participant describes choosing to work independently instead of collaborating	"So I just brought this yellow box by myself."
Behaviour	WT	Working Together	Participant describes choosing to collaborate with the agent	"We lift it together."
Behaviour	CS	Changing Strategy	Participant describes adapting their overall approach due to agent behaviour	"I just decided to move the box to myself because it could ignore my help and everything."