Research Article

Simulating Lay Health-Seeking Behavior with LLM Personas and Illness Vignettes: Reproducibility, Prompt Sensitivity, and Slice Dependence

Yuusuke Harada[1]

1. Hiroshima University, Japan

Large language models (LLMs) are increasingly used as "synthetic respondents" to simulate human judgments and decision-making. In healthcare-adjacent settings, a key methodological risk is that simulated behavior may be sensitive to prompt framing, run-to-run stochasticity, and the slice of scenarios being tested (e.g., red-flag vs non–red-flag situations). We present a fully synthetic, non-human-subject study that simulates a layperson persona choosing a next action when experiencing an illness vignette, using a fixed action codebook (A0–A9). In a Pilot experiment (40 persona–scenario pairs; 2 prompt variants; 3 repeats), the model produced plausibly monotonic action urgency as vignette severity increased and showed moderate run-to-run agreement (mean agreement 0.617). However, prompt comparisons performed within the same batch produced perfect agreement between prompts (0/40 mismatches), indicating that within-batch paired designs can underestimate prompt sensitivity. In an isolated-prompt audit (24 pairs), the action mismatch rate between prompts varied substantially across runs (0.0% to 45.8%). Prompt sensitivity was slice-dependent: mismatch was low in mild non–red-flag scenarios (8.3%) but high in red-flag scenarios (41.7%). A stress test using a stronger rubric shifted the action distribution (JS divergence 0.130) and reduced mean urgency by 1.29 points. These findings motivate multi-run, slice-aware evaluations when using LLM personas to simulate health-seeking behavior.

Corresponding author: Yuusuke Harada, yobou.hokkaido.harada@gmail.com

doi.org/10.32388/BEOZBC

1. Introduction

Simulating how “typical people” respond to illness symptoms is relevant to public health messaging, triage pathways, and behavioral research. Recent work in computational social science and psychology explores using LLMs as synthetic respondents to approximate human judgments, survey responses, and experimental results. This approach offers speed and scale, but also raises methodological risks: LLM outputs may be unstable across repeated runs, overly sensitive to prompt wording, and inconsistent across scenario slices.

In this Study, we focus on a narrow but operationally meaningful task: given (i) a synthetic layperson persona and (ii) a synthetic illness vignette, an LLM must choose a behavioral action category. Importantly, this study does not ask the model to provide medical advice; instead, it must choose among pre-defined action codes.

We empirically evaluate four aspects of methodological robustness: distribution plausibility (sanity checks), run-to-run repeatability across independent runs, prompt sensitivity measured as paired mismatch between prompt variants, and slice dependence of these effects across scenario strata such as red-flag vs non–red-flag.

2. Related work (selected)

LLMs as synthetic respondents (“silicon samples”) have been explored as proxies for subpopulations and as tools to replicate aspects of human-subject studies. At the same time, multiple papers warn that LLM-based synthetic data can diverge from real survey distributions and may flatten or misrepresent identity groups. Research on persona prompting and prompt robustness also highlights that seemingly small prompt differences can lead to large changes in outputs.

This study contributes a healthcare-adjacent, fully synthetic benchmark emphasizing prompt/run/slice interactions and a concrete evaluation protocol suited to solo execution and open materials.

3. Methods

3.1. Synthetic personas, scenarios, and codebook

We used 60 synthetic layperson personas (`personas.json`), 30 synthetic illness vignettes (`scenarios.json`), and a closed action/timing/reason schema (`codebook.json`).

code	description_en
A0	Watchful waiting (rest and monitor symptoms)
A1	Self-care (lifestyle adjustments: rest/hydration; no drug names)
A2	OTC medication / home remedies (consider OTC; no drug names/doses)
A3	Consult a pharmacy/pharmacist (visit or phone)
A4	Consult family/friends
A5	Search online for information (web/video/social media/government sites)
A6	Telephone consultation (health/medical hotline; no phone numbers)
A7	Schedule/visit a non-emergency clinic (primary care)
A8	Go to an emergency department (self-transport)
A9	Call emergency services (ambulance)

Action codebook (A0–A9).

code	description_en
now	Immediately (0-30 minutes)
hours	Within a few hours
within_24h	Within 24 hours
days	Within a few days (2-7 days)
undecided	Undecided / not sure

Timing codebook.

Each scenario includes a severity level (1–3) and a `red_flag` indicator (true/false). All personas and scenarios are fictional and contain no personal data.

3.2. Task definition and safety constraints

For each persona–scenario pair, the LLM outputs a single JSON object containing the fields `action_primary` (`A0-A9`), `action_time` (now/hours/within_24h/days/undecided), `escalation_if_worse` (A0–A9), `reason_tags` (1–3 tags drawn from a fixed schema), and `confidence` (an integer from 0 to 100).

The prompt explicitly forbids medical advice, diagnosis, treatment suggestions, medication names/doses, and emergency phone numbers. The model is instructed to output JSON only.

3.3. Prompt variants

We compare two prompt formulations. In P1, persona information is expressed as a short narrative paragraph. In P2, the same persona attributes are presented as a structured rubric (key–value style), while keeping the task definition, safety constraints, and output schema identical.

To make the difference concrete, below is an abbreviated illustrative excerpt (one synthetic persona and one mild vignette; line breaks added; content translated from the synthetic Japanese inputs). Full prompt templates and runner wrappers are provided in the Supplementary Materials.

P1 (narrative persona; abbreviated excerpt)

```
You are a simulated layperson. Follow safety rules: no diagnosis, no treatment advice, no medication names/doses, and no emergency phone numbers.
Persona (narrative): A person in their 20s living alone in an urban area with good healthcare access. Budget is tight and health literacy is low. They have a primary care doctor but tend to hesitate unless the situation feels serious.
Scenario: Weekday daytime. Since this morning you feel tired with a mild fever around 37.5°C. Appetite is slightly reduced but you can drink fluids. You have work tomorrow. What do you do next?
Return JSON only with fields: action_primary (A0-A9), action_time (now/hours/within_24h/days/undecided), escalation_if_worse (A0-A9), reason_tags (1-3 from the fixed schema), confidence (0-100).
```

P2 (structured persona; abbreviated excerpt)

```
You are a simulated layperson. Follow safety rules: no diagnosis, no treatment advice, no medication names/doses, and no emergency phone numbers.
Persona (rubric): age_group=20s; living=alone; access=urban_good; budget=tight; health_literacy=low; trust_in_healthcare=medium; has_primary_care_doctor=yes; anxiety=medium.
Scenario: Weekday daytime. Since this morning you feel tired with a mild fever around 37.5°C. Appetite is slightly reduced but you can drink fluids. You have work tomorrow. What do you do next?
Return JSON only with fields: action_primary (A0-A9), action_time (now/hours/within_24h/days/undecided), escalation_if_worse (A0-A9), reason_tags (1-3 from the fixed schema), confidence (0-100).
```

Output schema (identical across prompts; abbreviated example)

```
{"action_primary":"A2","action_time":"now","escalation_if_worse":"A7","reason_tags":
["uncertainty","work_time_constraint"],"confidence":55}
```

3.4. Experimental design

Pilot (same-batch paired prompts).

We sampled 40 persona–scenario pairs and executed two prompt variants (P1 and P2) with three independent repeats per prompt (three runs). In the Pilot, the two prompts were executed within the same batch execution context.

Total outputs: 240 (40 pairs × 2 prompts × 3 repeats).

Audit01 (isolated prompts, base24 pairs).

To test whether within-batch comparisons underestimate prompt sensitivity, we created an audit set of 24 pairs (severity=2, red_flag=false; 12 scenarios × 2 personas). We ran three independent replications (r1–r3) where P1 and P2 were executed in separate, isolated runs (paired only at analysis time).

Slice audits.

We additionally ran two slice audits: a severity=1 and red_flag=false slice (24 pairs), and a red_flag=true slice (24 pairs; 6 scenarios × 4 personas).

Stress test.

We compared baseline P2 outputs to a stronger rubric variant (**P2_strong**) on the base24 pairs to quantify distribution shift.

A protocol overview is shown in Figure 1.

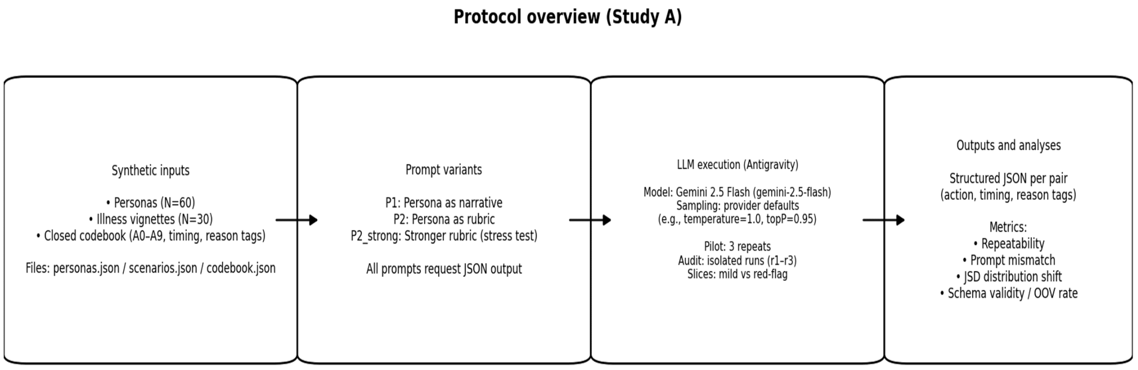
Protocol overview (Study A)
Synthetic inputs
• Personas (N=60)
• Illness vignettes (N=30)
• Closed codebook (A0–A9, timing, reason tags)
Files: personas.json / scenarios.json / codebook.json
Prompt variants
P1: Persona as narrative
P2: Persona as rubric
P2_strong: Stronger rubric (stress test)
All prompts request JSON output
LLM execution (Antigravity)
Model: Gemini 2.5 Flash (gemini-2.5-flash)
Sampling: provider defaults
(e.g., temperature=1.0, topP=0.95)
Pilot: 3 repeats
Audit: isolated runs (r1–r3)
Slices: mild vs red-flag
Outputs and analyses
Structured JSON per pair
(action, timing, reason tags)
Metrics:
• Repeatability
• Prompt mismatch
• JSD distribution shift
• Schema validity / OOV rate

Figure 1. *Protocol overview.*

3.5. Model, provider, and decoding configuration

All experiments were executed in the Antigravity environment using Google's Gemini 2.5 Flash model (model id: gemini-2.5-flash). We did not explicitly override the generation configuration in Antigravity for these runs; unless the platform applies task-specific overrides, requests therefore use provider-default sampling parameters. We did not retain platform-level request logs for these runs, so we cannot retrospectively verify the exact runtime decoding configuration; accordingly, we treat documented provider defaults as the best available reference and report them for transparency. For transparency, Vertex AI documentation for Gemini 2.5 Flash reports default values of temperature 1.0 and topP 0.95 (with topK fixed at 64 and candidateCount defaulting to 1)[1]. Because prompt sensitivity and repeatability can depend on these decoding parameters, future replications should explicitly set and record generation configuration and model version strings.

4. Metrics

4.1. Distribution plausibility (sanity checks)

We examine whether action urgency increases monotonically with scenario severity and with `red_flag=true`.

4.2. Repeatability across runs

For each (persona, scenario, prompt), we compute the proportion of runs agreeing with the modal `action_primary`. With three repeats (n=3), per-pair agreement is necessarily discrete and can take only the values 1/3, 2/3, or 1.0. We report mean agreement across pairs and summarize agreement by severity. Because we average across many pairs (N=40 in the Pilot; N=24 in the audit slices), the aggregate mean agreement is more stable than any single pair-level estimate; nevertheless, the small `n` limits how precisely per-pair reliability can be estimated and motivates larger repeat counts (e.g., 5–10) in future work when cost allows.

4.3. Prompt sensitivity (paired mismatch)

Within each run or slice, we compute the mismatch rate between P1 and P2 for `action_primary`, `action_time`, and `escalation_if_worse`.

4.4. Distribution shift (Jensen–Shannon divergence)

We quantify changes in the marginal action distribution using Jensen–Shannon divergence (base-2; bits) between two distributions (e.g., baseline vs stress-test rubric).

4.5. Output validity (schema adherence)

As a quality-control check, we compute an out-of-vocabulary (OOV) rate for `reason_tags` relative to the codebook.

4.6. Statistical tests (exploratory)

Because the study is framed as a methodological audit rather than a population estimate, our primary emphasis is on effect sizes (mismatch rates, agreement, and distribution shift). To address reviewer concerns about statistical support, we additionally report exploratory hypothesis tests. For run-to-run differences in mismatch on the same set of pairs (Audit01 r1–r3), we treat each persona–scenario pair as a matched unit and use Cochran’s Q test for more than two related samples, followed by pairwise exact McNemar tests with Holm correction for multiple comparisons. For comparisons between disjoint slices (for example, mild vs red-flag scenarios), we use Fisher’s exact test on a 2×2 table of mismatches vs matches. For the stress test, we treat action codes as an ordinal urgency score (A0=0 … A9=9) and compare baseline vs strong rubric using a paired Wilcoxon signed-rank test. All tests are two-sided and intended as descriptive complements to the reported effect sizes. Because the number of pairwise comparisons in Audit01 is small (three run pairs), Holm adjustment is mild; our qualitative conclusions are unchanged if unadjusted p-values are considered.

5. Results

5.1. Pilot: distribution plausibility

Action urgency increased with vignette severity (Figure 2) and was higher when red_flag=true (Figure 3).

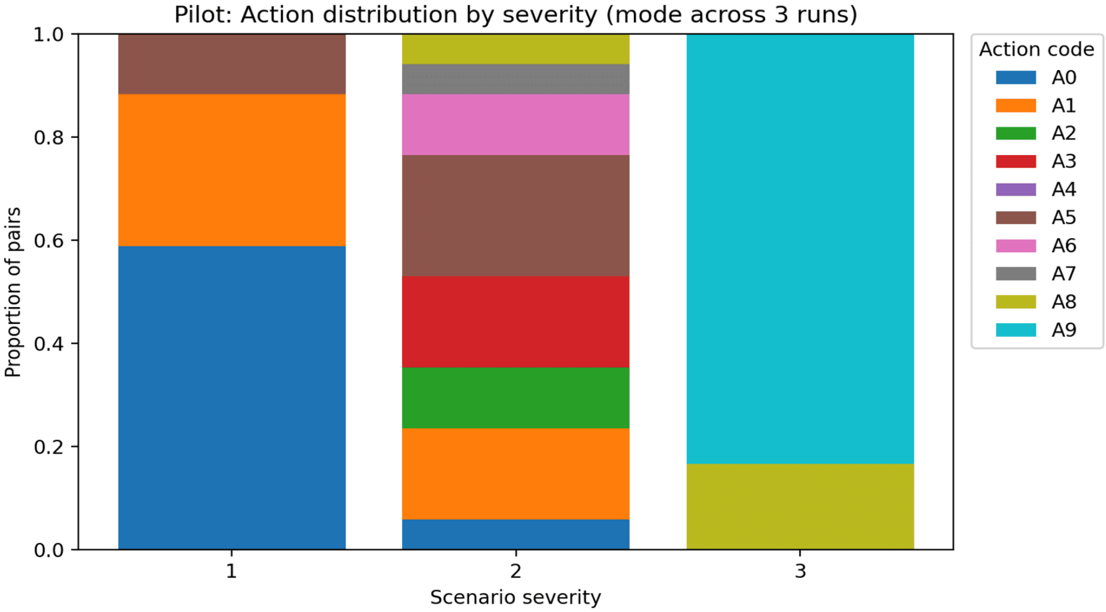

Figure 2. *Pilot action distribution by severity (mode across 3 runs).*

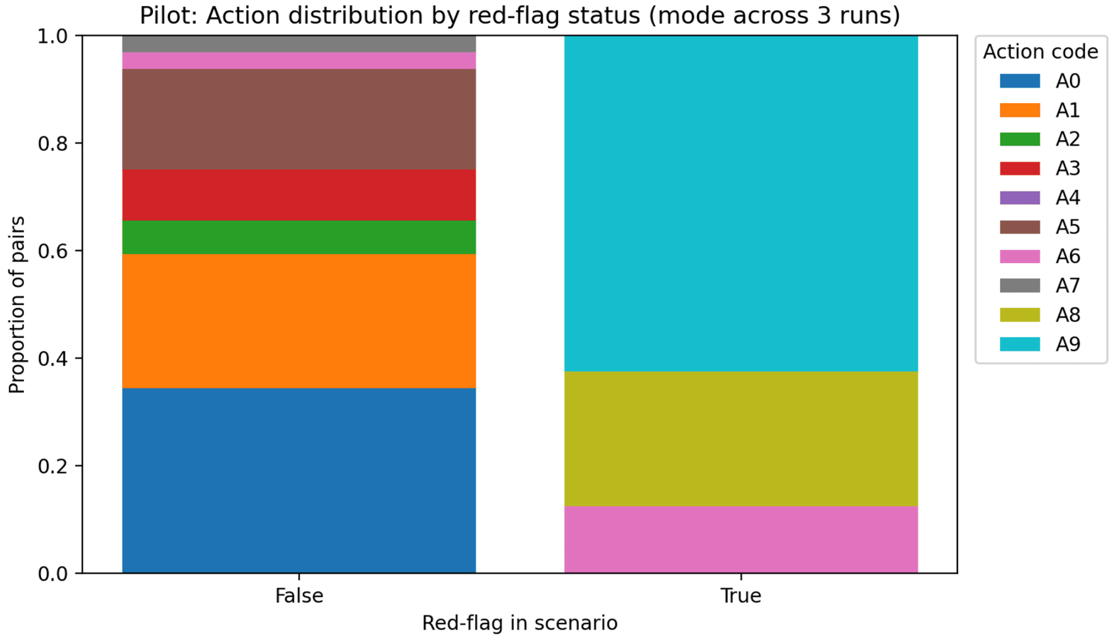

Figure 3. *Pilot action distribution by red-flag indicator (mode across 3 runs).*

5.2. Pilot: repeatability

Across the 40 pairs, mean run-to-run agreement for action_primary was 0.617 (Figure 4). When summarized across pairs, the standard error of the mean agreement was approximately 0.041, reflecting that we average many coarse per-pair estimates (each based on n=3 runs). Agreement differed by severity, with higher repeatability for severity=3.

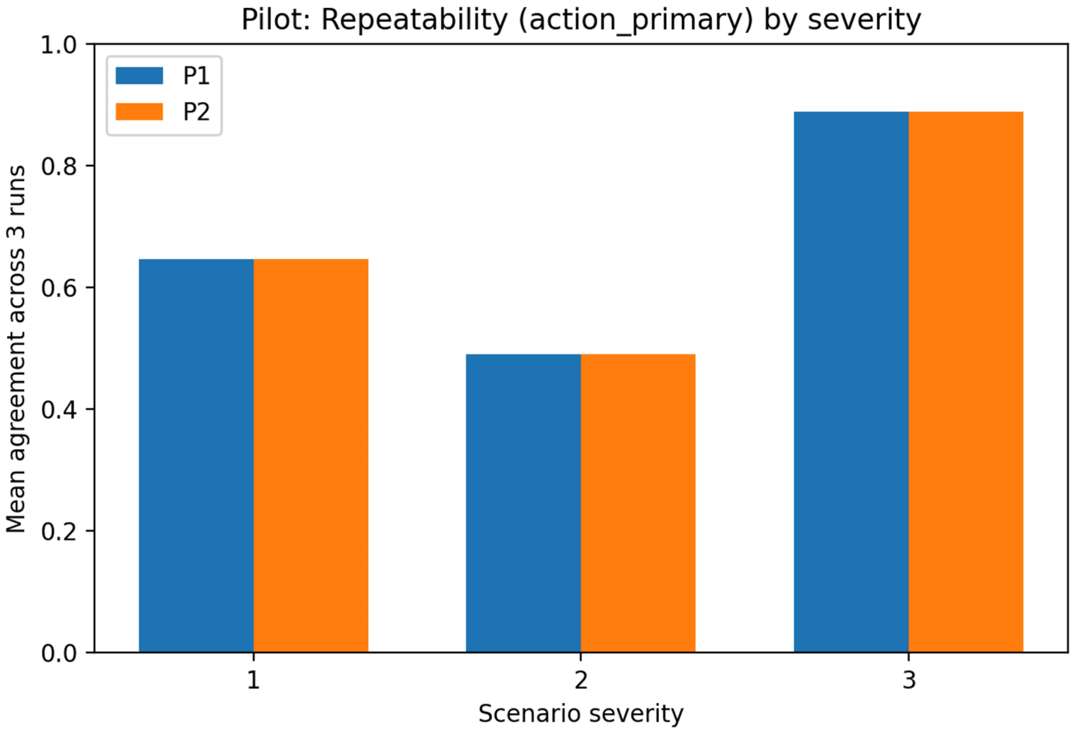

Figure 4. *Pilot repeatability by severity (mean agreement across 3 runs).*

5.3. Pilot: perfect prompt agreement within batch (a cautionary result)

In the Pilot, P1 and P2 produced identical outputs for action_primary on all 40 pairs when compared within the same run/batch (0 mismatches). Because the prompts were executed in the same batch context, this design may underestimate prompt sensitivity (e.g., due to within-batch coupling or latent reuse of earlier outputs).

5.4. Audit01: prompt mismatch varies strongly across runs

In the isolated-prompt Audit01 (base24 pairs), the P1 vs P2 mismatch rate for `action_primary` varied widely across replications, ranging from 0.0% (r1) to 45.8% (r3), with an intermediate value of 8.3% (r2).

To quantify whether this heterogeneity is larger than expected from sampling noise alone, we performed a paired Cochran's Q test on per-pair mismatch indicators across r1–r3. The test rejects equality across runs (Q=17.17, df=2, p=0.00019). Pairwise exact McNemar tests (Holm-adjusted) indicate that r3 differs from r1 (p=0.00293) and from r2 (p=0.0234), whereas r1 and r2 are not distinguishable (p=0.50). A detailed summary of tests is provided as Supplementary Table S1 (statistical_tests.csv).

Mismatch rates for multiple fields and slices are shown in Figure 5.

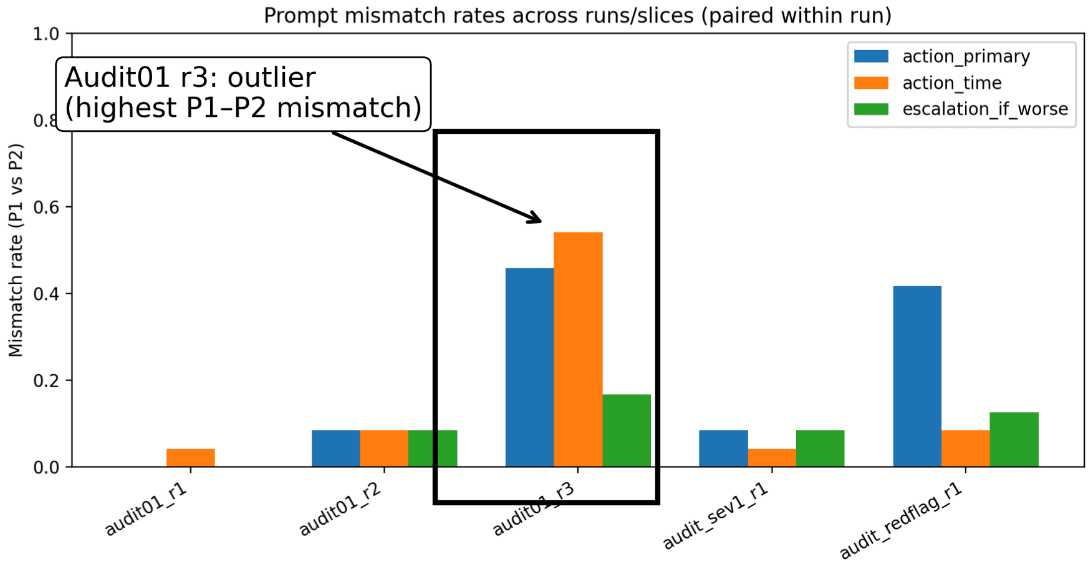

Figure 5. *Prompt mismatch rates across runs and slices (paired within run). Annotation highlights Audit01 r3 as an outlier run.*

5.5. Slice dependence: mild vs red-flag scenarios

Prompt mismatch for action_primary was low in the severity=1 & red_flag=false slice (8.3%; 2/24 pairs) but high in the red_flag=true slice (41.7%; 10/24 pairs). A Fisher exact test comparing these mismatch counts rejects equality (odds ratio=0.127, p=0.01734), indicating significantly higher prompt sensitivity in the red-flag slice. This suggests that prompt sensitivity is not constant: it can depend on scenario strata that are especially important for safety.

5.6. Audit01: repeatability across runs differs by prompt

Across the three Audit01 replications, mean agreement for action_primary was 0.625 for P1 and 0.764 for P2 (Figure 6).

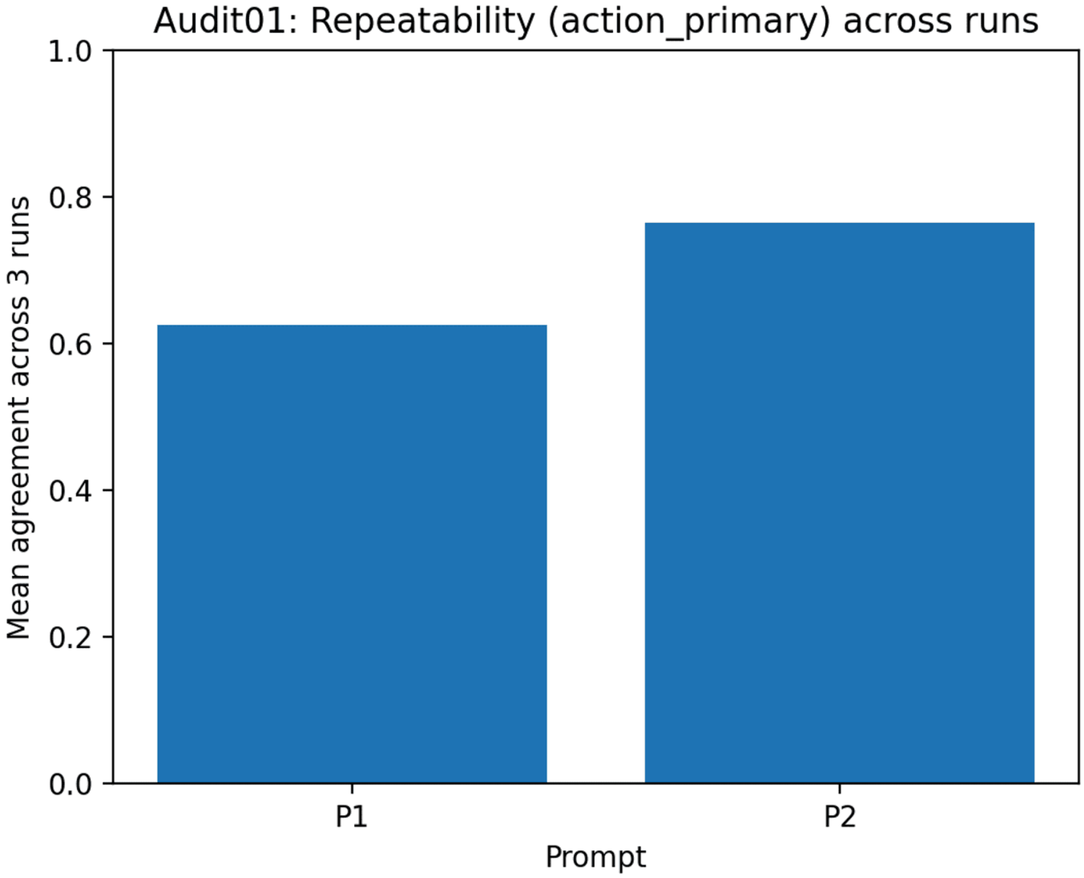

Figure 6. *Audit01 repeatability across runs (mean agreement for action_primary).*

5.7. Stress test: stronger rubric shifts distributions

Compared to baseline P2, the stronger rubric variant (P2_strong) produced substantial action distribution shift (Figure 7), with Jensen–Shannon divergence (base-2) of 0.130. Because JS divergence is bounded between 0 (identical distributions) and 1 (maximally separated distributions) in the base-2 convention, 0.130 represents a moderate shift in a 10-category action distribution[2]. Concretely, relative to baseline, P2_strong reduced clinic seeking (A7: 5→1 of 24) and information seeking (A5: 6→3) while increasing lower-urgency actions such as OTC/home remedies and pharmacy consultation (A2: 2→6; A3: 1→3). Treating action codes as an ordinal urgency score (A0=0…A9=9), mean urgency decreased by 1.29 points (3.54→2.25), and a paired Wilcoxon signed-rank test indicates that this reduction is statistically detectable (W=0.0, p=0.00484).

For reference, prompt-induced action-distribution JS divergence between P1 and P2 ranged from 0.022 in the mild slice to 0.140 in Audit01 r3, suggesting that rubric-strengthening can be comparable in magnitude to large prompt effects observed under isolation.

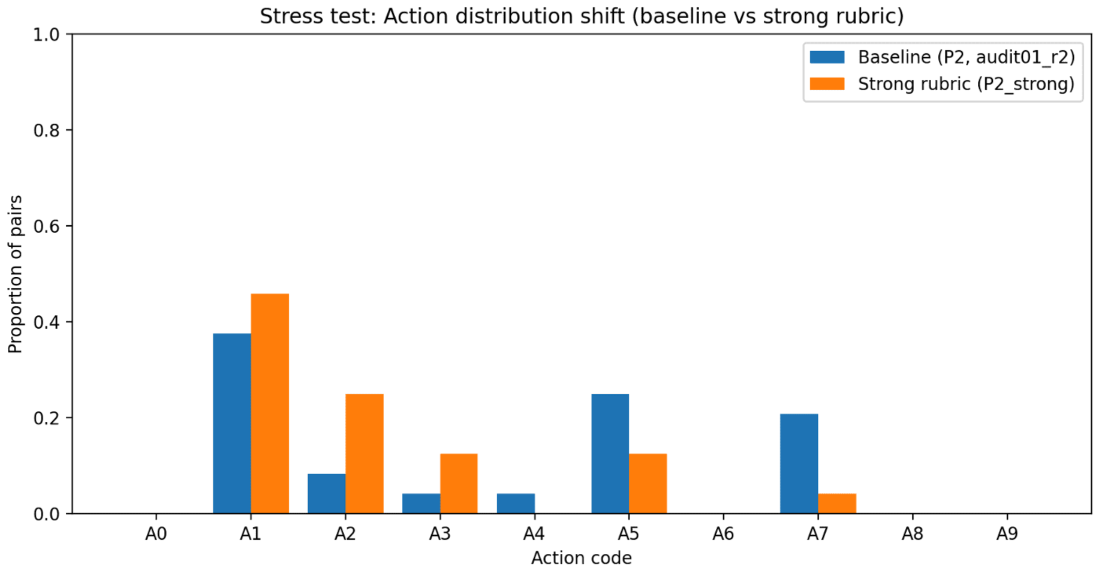

Figure 7. *Stress test action distribution shift (baseline vs strong rubric).*

5.8. Schema adherence: reason_tags OOV drift in some runs

While the Pilot and Audit01 r1 adhered to the reason_tags schema, several subsequent audit and slice runs produced semantically plausible but schema-invalid tags (high OOV rates in Figure 8). In practice, the model often replaced codebook keys with natural synonyms (for example, producing tags such as “fear” or “severity_perception” instead of selecting the exact predefined keys). This pattern suggests that OOV inflation can arise both from the complexity of simultaneously satisfying multiple constraints (choose an action, choose timing, choose 1–3 tags, and remain safety-compliant) and from the model’s tendency to prioritize semantic adequacy over exact-string compliance. For downstream quantitative analysis, this is a critical failure mode: apparent “reasons” may drift out of the controlled vocabulary even when the action codebook is stable. A practical mitigation is to enforce structured output using a formal JSON schema with enumerated allowed values for reason_tags, combined with strict validation and retry logic at the pipeline level[3].

doi.org/10.32388/BEOZBC

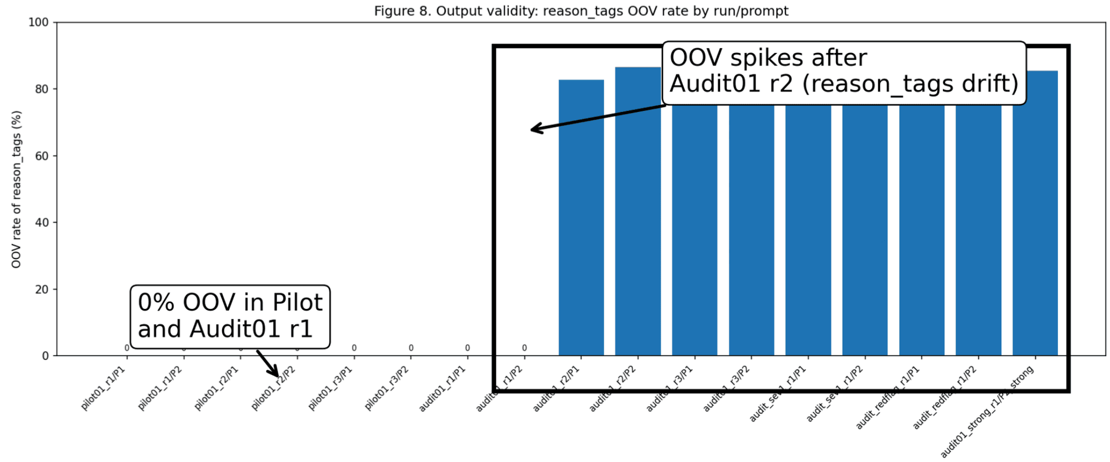

Figure 8. *Output validity: reason_tags OOV rate by run/prompt. Annotation highlights the step-change from 0% OOV (Pilot, Audit01 r1) to high OOV in subsequent audits.*

5.9. Summary table of key results

Finding	Value
Pilot run-to-run agreement (action_primary)	0.617 (mean across pairs; 3 runs)
Pilot prompt mismatch P1 vs P2 (action_primary)	0/40 pairs (identical within batch)
Audit01 prompt mismatch (action_primary)	r1: 0.0%, r2: 8.3%, r3: 45.8%
Audit01 repeatability across runs (action_primary)	P1: 0.625, P2: 0.764
Slice dependence (action_primary mismatch)	sev1/rfFalse: 8.3%, red_flag True: 41.7%
Stress test (P2_strong vs baseline P2)	mismatch: 41.7%, JS divergence: 0.130, mean urgency shift: -1.29

Table 1.

doi.org/10.32388/BE0ZBC

5.10. External plausibility check against public triage guidance (illustrative, non-ground-truth)

Study is designed as a methodological audit rather than a clinical validity study, and it does not claim that simulated actions reflect real patient behavior. Nevertheless, readers may reasonably ask whether the action codes produced by the LLM remain within a broadly "common-sense" triage range. As a small, illustrative plausibility check (not ground truth), we compared a subset of red-flag vignettes to publicly available triage guidance. These sources generally recommend urgent emergency evaluation for sudden severe breathing difficulty with chest pain, signs of anaphylaxis (for example, facial/lip swelling with throat tightness), and sudden extremely severe headache. In the red-flag slice, the model most often selected high-urgency actions (A8/A9) for these vignettes, especially under P2 (Table 2). This check should not be interpreted as medical advice and does not replace validation against human judgments, but it helps contextualize the output space.

scenario_id	Symptom summary (translated from vignette)	Example public triage guidance (summary)	Modal action (red-flag slice)
S008	Sudden, unusually severe headache ("worst so far"), with nausea	Public guidance commonly treats sudden extremely severe headache as an emergency presentation requiring urgent evaluation.[4]	P1: A8; P2: A8
S010	Severe shortness of breath and central chest pain, with anxiety and sweating	Public guidance recommends immediate emergency care for severe breathing difficulty and/or chest pain with shortness of breath.[5][6]	P1: A8; P2: A9
S018	Facial/lip swelling and throat discomfort, worsening over 30 minutes	Public guidance treats sudden swelling of lips/mouth/throat and breathing/swallowing difficulty as anaphylaxis requiring emergency response. [7]	P1: A8; P2: A8

Table 2. Illustrative comparison between selected red-flag vignettes and public triage guidance.

doi.org/10.32388/BE0ZBC

6. Discussion

The experiments in Study are intentionally modest in scale, but they expose a methodological issue that can materially affect how researchers interpret LLM-based simulations: within-batch paired prompt comparisons can substantially underestimate prompt sensitivity. In the Pilot, P1 and P2 produced perfect agreement when executed in the same batch context, which could easily be misread as evidence that "prompt formulation does not matter." However, in that setup the two conditions were generated within a shared response context, so the later segments were conditioned on earlier generated tokens. For autoregressive LLMs, this provides a direct coupling channel: later completions can reuse, imitate, or anchor to earlier completions simply because they appear in-context. This is closely related to in-context learning and formatting effects, where models infer latent "task rules" from nearby examples and reuse them across the sequence[8][9]. It can occur even when requests are otherwise "stateless" at the API level. Gemini endpoints explicitly support both stateful and stateless multi-turn interactions, but in either mode the conversation history (or earlier generated text) becomes part of the conditioning context for later outputs[10]. In addition, some providers offer context caching features that reuse previously processed prompt tokens (e.g., key–value states) for cost/latency reasons. In our setup we did not explicitly configure or reference a cache, but depending on the access route, implicit caching may still occur at the serving layer; therefore, simulation studies should report whether caching is enabled and whether cache keys are shared across conditions[11][12]. The isolated-run Audit01 contradicts the naive "prompt does not matter" interpretation, showing that the same P1–P2 comparison can range from no mismatches to nearly half of cases mismatching across different replications. Regardless of the precise mixture of mechanisms, the empirical implication is the same: paired designs that do not enforce independence can produce overconfident conclusions about robustness.

The slice audits highlight a second layer: prompt sensitivity is not constant across the scenario distribution. We observed substantially higher mismatch in red-flag scenarios than in mild non–red-flag scenarios. This is consistent with the idea that red-flag vignettes sit closer to a decision boundary where multiple actions may be defensible; in such boundary cases, small changes in how persona attributes are framed can tip the model from one action category to another. For healthcare-adjacent simulations, this matters because red-flag slices correspond to high-stakes decision regions, and errors or instability there have disproportionate interpretive impact. Practically, this suggests that reporting only an average mismatch rate over a mixed scenario set can mask important conditional failure modes.

The stress test illustrates that “making prompts more structured” is not a neutral change. Strengthening the rubric (P2_strong) shifted the overall action distribution and reduced average urgency, which could be viewed as either an improvement or a bias depending on the intended construct. This reinforces the need to treat prompt design as part of the measurement instrument. In survey methodology terms, prompt variants are alternative instruments that can induce systematic measurement error; therefore, prompt changes should be audited in the same way as changes to a questionnaire.

Finally, schema adherence results underscore that even when a controlled vocabulary is provided, models may drift toward semantically adequate but schema-invalid outputs unless constraints are enforced. In our runs, action_primary remained largely within the closed codebook, while reason_tags showed substantial OOV drift in several conditions. One interpretation is that the action codebook is cognitively simpler and more salient to the model than the reason-tag vocabulary, and that the prompt’s multi-objective structure makes strict lexical compliance brittle. For the broader research program of “synthetic respondents,” this has two implications. First, reproducibility cannot be assessed only at the level of high-level outcomes; auxiliary fields that are intended to support interpretation (such as reasons) can silently degrade. Second, structured-output mechanisms that accept an explicit JSON Schema / response schema (including enumerated string fields) are available for Gemini endpoints and provide a direct technical path to eliminating OOV drift by construction[3][13][14]. Under enum-constrained schemas, we expect OOV rates to approach zero, but the constraint can still shift marginal distributions by collapsing synonyms or changing how the model handles uncertainty, so effects on prompt sensitivity should be measured rather than assumed. Moreover, schema enforcement is not purely a formatting change: it can alter the distribution of outputs by constraining the feasible set, and Vertex AI documentation notes that structured output can reduce model quality in some settings; therefore, it should be audited as a first-class experimental factor rather than assumed to be neutral[14].

An important nuance is that “fixing” schema adherence is not purely a post-processing choice: enforcing a closed JSON Schema (e.g., an enum over reason_tags) changes the generation problem itself. When the model’s preferred surface form is outside the allowed vocabulary, a strict constraint forces the output into the nearest admissible label, which can reduce OOV while also altering the marginal distribution of tags. In extreme cases, this can produce apparent gains in repeatability simply by collapsing many paraphrases into a small label set, while masking semantic disagreement or uncertainty.

Moreover, because reason_tags are part of the prompt–output contract, constraining them may have second-order effects. Some prompt designs encourage the model to “think through” tags before committing to an action; if the tag space is restricted, the model's latent rationale may be nudged toward a different explanation, potentially shifting action selections as well. For this reason, future work should treat schema enforcement as an experimental factor rather than a neutral engineering fix: run both unconstrained and constrained variants, quantify any induced distribution shift (e.g., JSD), and evaluate whether lower OOV corresponds to improved semantic validity or merely to forced compliance.

Operationally, there are multiple implementation paths—guided decoding against a schema, iterative self-repair ("please output valid JSON"), and post-hoc normalization that maps synonyms to canonical tags. Each approach trades off strict validity against potential bias and should be audited explicitly, especially when downstream conclusions depend on the frequency of specific tags or on comparisons between prompts/runs.

Decoding configuration deserves similar treatment. Temperature and nucleus sampling (topP) determine how much stochastic exploration the model performs when generating a categorical choice. Vertex AI documentation explains that temperature is applied during sampling (when topP/topK are used) and that temperature 0 corresponds to always selecting the highest-probability token, producing responses that are mostly deterministic[15]. In a closed-set decision task, raising temperature mainly affects boundary cases and can increase both run-to-run variance and apparent prompt sensitivity, because small changes in persona framing interact with stochastic tie-breaking. Practitioners can further reduce stochasticity by lowering topP/topK and, where available, by setting an explicit random seed on requests; documentation notes this is best-effort and deterministic output is not guaranteed[15]. For practical use, we suggest reporting results at two operating points: a near-deterministic baseline (e.g., temperature 0–0.2 with default topP) that supports reproducible instrument checks, and a stochastic setting (e.g., the model's default temperature 1.0) that better captures the distributional variability relevant to simulation[1]. Deterministic decoding can improve repeatability, but it can also hide uncertainty by collapsing borderline cases into a single mode; therefore, for simulation studies that interpret distributions, explicit multi-run sampling at a moderate temperature remains important.

Taken together, these findings motivate a conservative evaluation practice for LLM-based behavior simulation: isolate conditions when comparing prompts, replicate across multiple runs, stratify analyses by scenario slices, and report model identifiers and decoding parameters. Even when the ultimate goal is substantive, methodological audits of this form are a prerequisite for interpreting synthetic outputs responsibly.

#7. Limitations and future work

This is a single-model study conducted with one model family, and results may differ across model providers, versions, or access routes; multi-model evaluation remains an important next step. Because the study is fully synthetic, we do not claim that the simulated distributions match real populations, and we treat the experiments as a methodological audit rather than a behavioral estimate. We also do not provide human labels or expert ground truth for the “correct” action codes; the small comparison to public triage guidance in Section 5.10 is illustrative only and should not be interpreted as validation. The scenario space is finite and synthetic; expanding coverage and incorporating expert-designed or empirically grounded vignettes would strengthen external validity.

The Pilot uses three repeats per condition (n=3) to estimate run-to-run variability in a feasible solo workflow. This choice is sufficient to detect gross instabilities and to map which slices are vulnerable, but it yields coarse per-item repeatability estimates because agreement values are quantized in 1/3 increments. If the goal is to estimate reliability for a specific persona–scenario pair or to report tight confidence intervals for repeatability within a slice, larger repeat counts (e.g., 5–10) are more appropriate, and sequential designs that allocate extra repeats to the most variable slices can control cost while improving precision.

The statistical tests reported in Section 4.6 are exploratory complements to effect sizes; they treat persona–scenario pairs as analysis units and do not eliminate all potential dependence structures, so p-values should be interpreted cautiously. We did not systematically sweep decoding parameters (temperature, topP) or model versions, and these settings can directly affect both repeatability and prompt sensitivity; a principled extension is to treat decoding as another factor, report results at multiple temperatures, and separate stochastic-decoding variance from prompt variance. Similarly, although we discuss structured outputs and schema enforcement, our main runs relied on prompt instructions and post-hoc validation rather than enforcing a response schema at generation time. Future work should directly compare unconstrained prompting versus JSON Schema–constrained structured output to quantify how much OOV drift is eliminated and whether constraint enforcement changes action distributions or mismatch rates.

8. Conclusion

Using LLM personas to simulate lay health-seeking behavior is feasible in a fully synthetic, non-human-subject setting and can yield plausible distributions. However, prompt sensitivity can be underestimated by within-batch designs and can vary across runs and scenario slices. We recommend multi-run, isolated-prompt evaluations with explicit slice audits and schema validation as a minimum standard for this research direction.

Statements and Declarations

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests

The author declares no competing interests.

Ethics approval and consent to participate

Not applicable. This study used only fully synthetic personas and fictional illness vignettes; no human participants, patient records, or identifiable data were involved.

Consent for publication

Not applicable.

Availability of data and materials

All synthetic inputs, model outputs, and analysis scripts needed to reproduce the reported quantitative results are provided in the Supplementary Materials archive.

Author contributions

YH conceived the study, designed the experiments, implemented the pipeline, analyzed the data, and wrote the manuscript.

Use of generative AI tools

Generative AI was used as the experimental object of study and to support editing for clarity. All analyses, interpretations, and final decisions about content were made by the author.

References

1. [a, b] *Google Cloud (2025). "Gemini 2.5 Flash (Vertex AI Model Documentation)." Google Cloud. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash.*
2. [^] *Lin J (1991). "Divergence Measures Based on the Shannon Entropy." IEEE Trans Inf Theory.* **37***(1):145–151. doi:10.1109/18.61115.*
3. [a, b] *Google AI for Developers (2026). "Structured Outputs (Gemini API Documentation)." Google AI for Developers. https://ai.google.dev/gemini-api/docs/structured-output.*
4. [^] *University Hospitals Sussex NHS Foundation Trust (2025). "Headaches: Emergency Department Patient Leaflet." University Hospitals Sussex NHS Foundation Trust. https://www.uhsussex.nhs.uk/resources/headaches-emergency-department-patient-leaflet/.*
5. [^] *NHS (2026). "Chest Pain." NHS. https://www.nhs.uk/symptoms/chest-pain/.*
6. [^] *NHS (2026). "Shortness of Breath." NHS. https://www.nhs.uk/symptoms/shortness-of-breath/.*
7. [^] *NHS (2026). "Anaphylaxis." NHS. https://www.nhs.uk/conditions/anaphylaxis/.*
8. [^] *Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020). "Language Models Are Few-Shot Learners." Advances in Neural Information Processing Systems (NeurIPS 2020). https://arxiv.org/abs/2005.14165.*
9. [^] *Min S, Lyu X, Holtzman A, Artetxe M, Lewis M, Hajishirzi H, Zettlemoyer L (2022). "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" Proceedings of EMNLP 2022. https://aclanthology.org/2022.emnlp-main.759/.*
10. [^] *Google AI for Developers (2025). "Interactions API | Gemini API (Stateful Vs Stateless Conversations)." Google AI for Developers. https://ai.google.dev/gemini-api/docs/interactions.*
11. [^] *Google Cloud (2025). "Context Caching Overview | Generative AI on Vertex AI." Google Cloud. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/context-cache/context-cache-overview.*
12. [^] *Google Cloud (2025). "Use Vertex AI Context Caching With Gemini to Save Costs and Reduce Latency." Google Cloud Blog. https://cloud.google.com/blog/products/ai-machine-learning/vertex-ai-context-caching.*
13. [^] *Google AI for Developers (2026). "Models (Gemini API Documentation)." Google AI for Developers. https://ai.google.dev/gemini-api/docs/models.*

14. [a, b] *Google Cloud (2026). "Structured Output | Generative AI on Vertex AI." Google Cloud. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/multimodal/control-generated-output.*

15. [a, b] *Google Cloud (2026). "Content Generation Parameters | Generative AI on Vertex AI." Google Cloud. https://docs.cloud.google.com/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters.*

Supplementary data: available at https://doi.org/10.32388/BE0ZBC

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.