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, stochastic decoding, and the scenario slice being tested (e.g., red-flag vs non–red-flag situations). We present a fully synthetic, non-human-subject methodological audit in which an LLM, conditioned on a fictional layperson persona, selects a next-action code (A0–A9) for an illness vignette. In a Pilot experiment (40 persona–scenario pairs; two prompt variants; three repeats), action urgency increased with vignette severity and repeatability was moderate (mean modal agreement 0.617). However, within-batch paired prompt comparisons yielded perfect agreement (0/40 mismatches), suggesting that paired designs that do not enforce independence can severely underestimate prompt sensitivity. In an isolated-prompt audit (24 pairs; three repeats), prompt mismatch varied widely across replications (0.0% to 45.8%). To disentangle prompt effects from decoding noise, we performed a controlled follow-up rerun on two slices (non–red-flag and red-flag; 24 pairs each) under explicit decoding settings (temperature 0 vs default temperature 1.0/top-p 0.95) with 8–10 repeats per condition. Prompt sensitivity remained high under both decoding regimes (mean action mismatch 0.787–0.821; mean Jensen–Shannon divergence 0.148–0.196 bits), while near-deterministic decoding improved run-to-run stability in three of four settings. A rubric stress test shifted the action distribution (Jensen–Shannon divergence 0.130) and reduced mean urgency by 1.29 points. Together, these results motivate multi-run, slice-aware, decoding-controlled evaluations when using LLM personas for behavioral simulation.

 doi.org/10.32388/BEoZBC.2

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

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* (e.g., self-care, consult a pharmacist, visit a clinic, call emergency services). 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 have increasingly been used as "synthetic respondents" or "silicon samples" to approximate human distributions, explore subgroup opinion, and replicate aspects of human-subject studies[1][2][3][4][5]. Related work has also examined persona and socio-demographic prompting as mechanisms for conditioning these synthetic samples[6][7], and agent-based simulations that generate interactive human-like behavior traces[8]. In the health domain, vignette-driven simulation of patient populations with LLMs has been explored as a means to generate diverse synthetic cases[9].

At the same time, several lines of scholarship caution against treating LLM outputs as direct replacements for human judgments. Critiques emphasize the risks of misrepresentation, identity flattening, and ecological mismatch between synthetic and real survey distributions[10][11][12][13]. Empirical analyses further show that model outputs can reflect the distribution of opinions present in training data rather than any "average human" and can encode systematic response biases[14][5]. Related

behavioral benchmarking work (e.g., theory-of-mind tasks) similarly highlights sensitivity to prompting and evaluation design, underscoring the need for careful experimental controls (Kosinski, 2024; Strachan et al., 2024). Finally, work on prompt robustness and benchmarking frameworks highlights that small prompt changes can materially shift outcomes, motivating standardized, auditable evaluation pipelines[15].

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 (paired within run).

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.

Controlled follow-up (decoding-controlled reruns). To address reviewer concerns about uncontrolled generation configuration, we reran two key slices (audit01_base24 and redflag24; 24 pairs each) under two explicitly specified decoding configurations: a near-deterministic configuration det_t0 (temperature 0.0, top_p 1.0, top_k 20, candidate_count 1, max_output_tokens 512, fixed seed 17) and a stochastic configuration stoch_default (temperature 1.0, top_p 0.95, top_k 20, candidate_count 1, max_output_tokens 512, per-run seeds). For each slice, we executed P1 and P2 in per-item isolated mode (one request per persona–scenario item) and targeted 10 repeats per condition; completed repeats ranged from 8 to 10 due to occasional transient failures.

A protocol overview is shown in Figure 1.

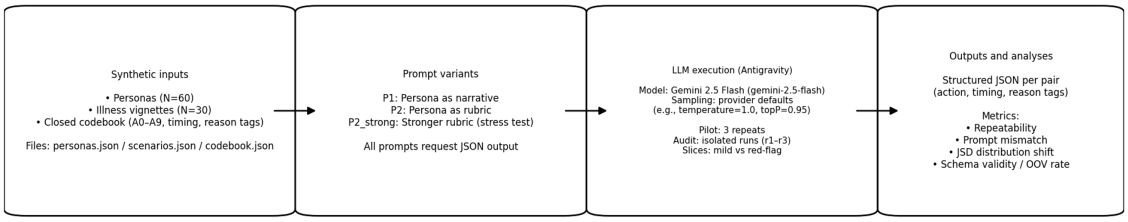

Figure 1. *Protocol overview.*

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) via Vertex AI. For the initial Pilot/audit/stress-test runs, we did not explicitly override the generation configuration in Antigravity and did not retain platform-level request logs; therefore, we treat documented provider defaults as the best available reference (temperature 1.0, topP 0.95; candidateCount 1; topK as documented)[16][17]. In response to reviewer feedback, we additionally conducted a controlled follow-up in which we explicitly set and recorded decoding parameters in the API call. We used two configurations: det_t0 (temperature 0.0, top_p 1.0, top_k 20, candidate_count 1, max_output_tokens 512, fixed seed 17) and stoch_default (temperature 1.0, top_p 0.95, top_k 20, candidate_count 1, max_output_tokens 512, per-run seeds). Unless otherwise stated, all experiments use single-turn, stateless requests with candidate_count=1. We did not enable explicit context caching features, but we discuss potential within-batch coupling and serving-layer confounds in Section 6.

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 (modal agreement). 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 therefore interpret n=3 primarily as a low-cost screen for

gross instability rather than a precise reliability estimate. 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. In the controlled follow-up, we increase repeats to 8–10 per condition, producing less-quantized agreement estimates and enabling direct comparisons of repeatability under different decoding configurations.

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 prompt 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. For the controlled decoding follow-up, we compare per-pair modal agreement under det_t0 versus stoch_default using paired Wilcoxon signed-rank tests (one-sided alternative det_t0 > stoch_default) within each slice and prompt; because this yields four related tests (two slices × two prompts), we report Holm-adjusted p-values. All

tests are intended as descriptive complements to the reported effect sizes, and p-values should be interpreted cautiously given potential dependence structures and the study's audit framing.

5. Results

5.1. Pilot: distribution plausibility

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

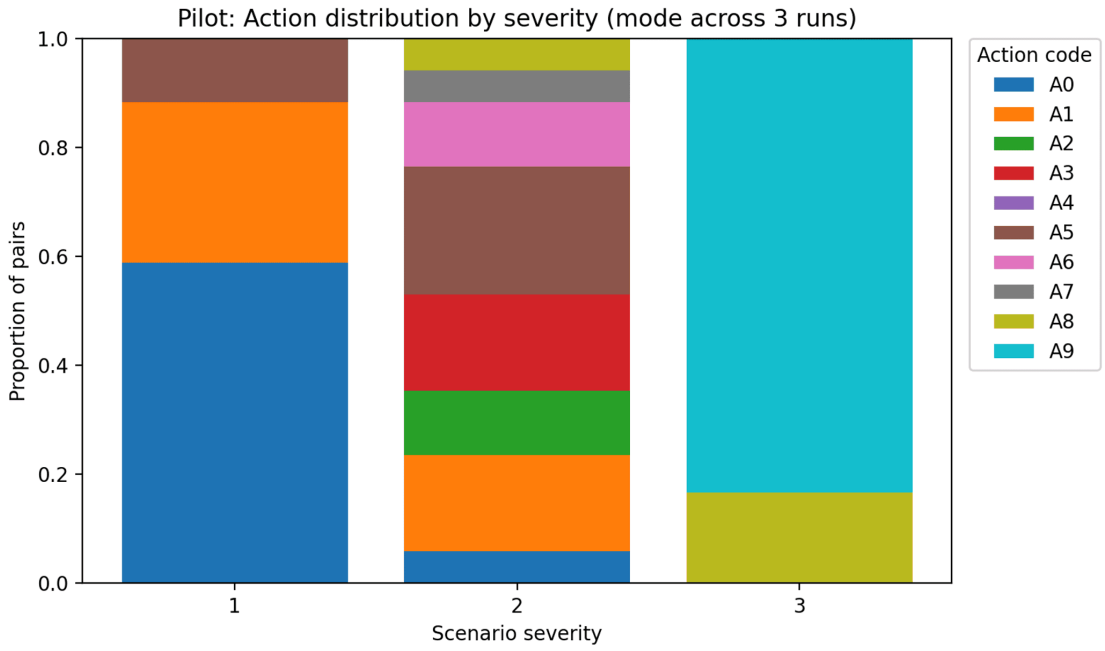

Figure 2. *Pilot action distribution by severity (mode across 3 runs).*

doi.org/10.32388/BEoZBC.2

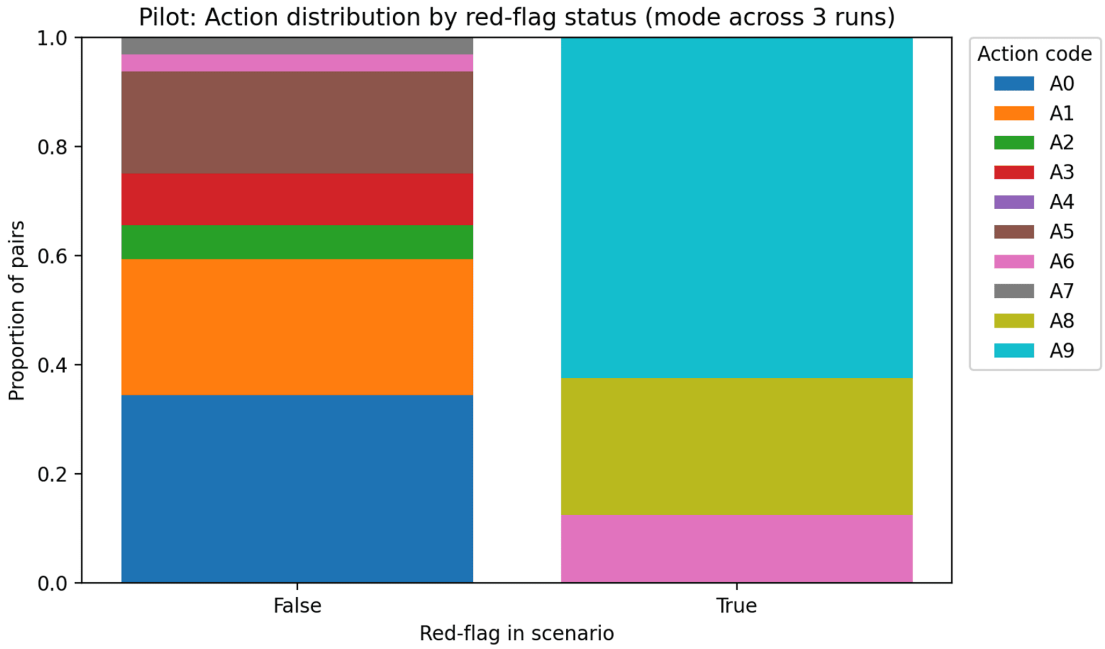

Figure 3. *Pilot action distribution by red-flag indicator (mode across 3 runs).*

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.

doi.org/10.32388/BEoZBC.2

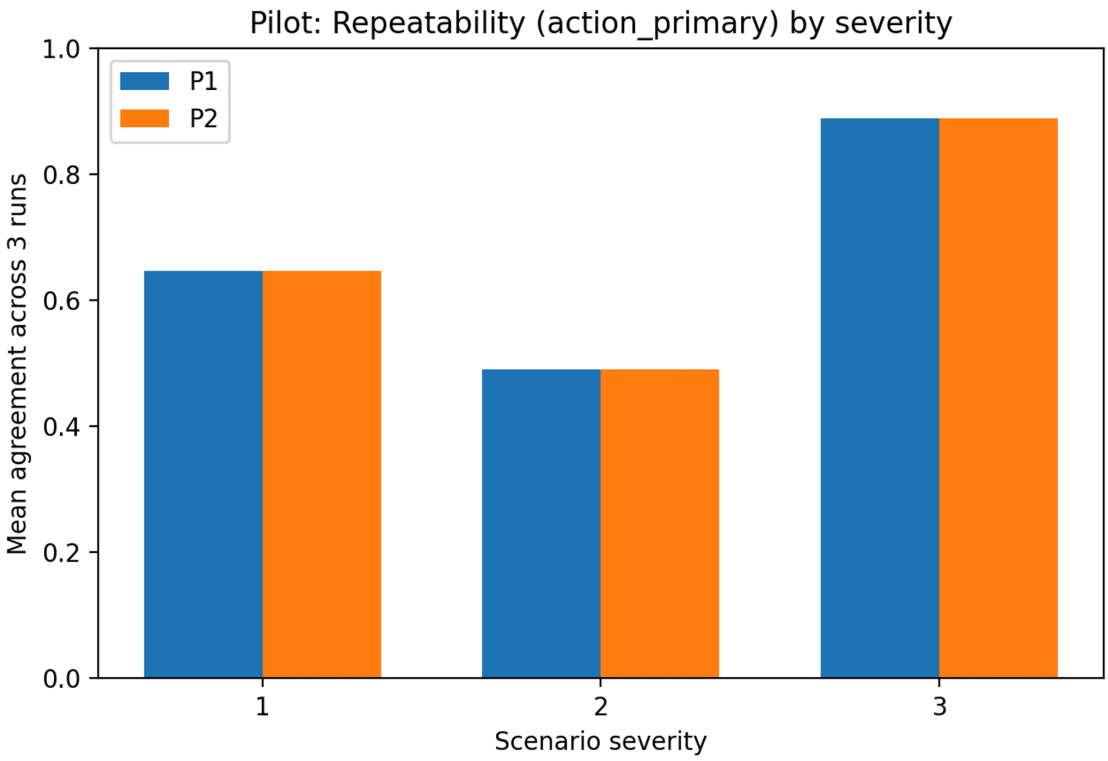

Figure 4. *Pilot repeatability by severity (mean agreement across 3 runs).*

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.

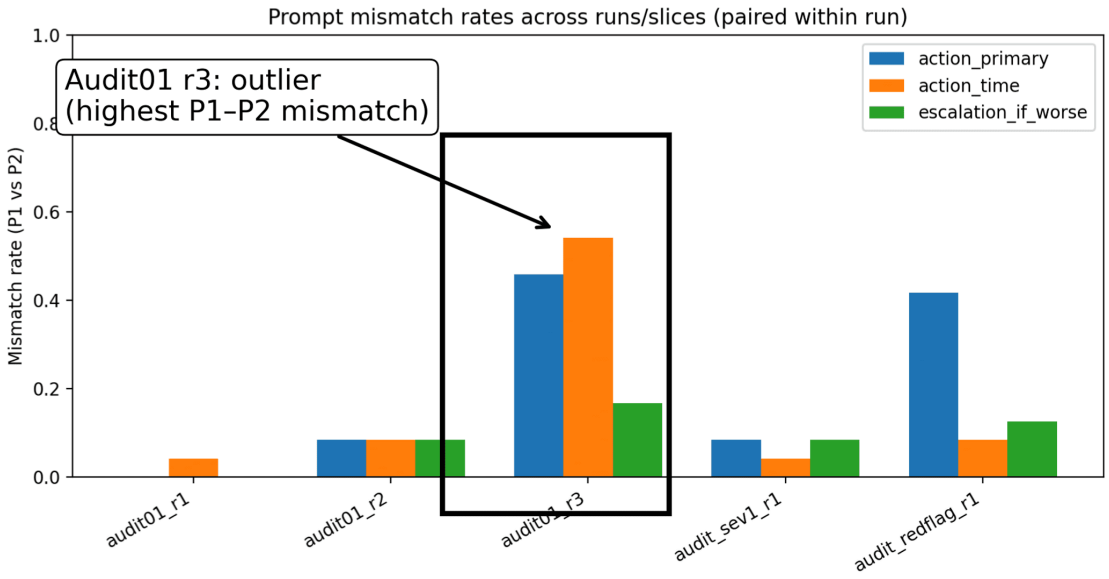

Figure 5. *Prompt mismatch rates across runs and slices (paired within run). Annotation highlights Audit01 r3 as an outlier run.*

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. A decoding-controlled rerun with higher repeat counts is reported in Section 5.7.

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).

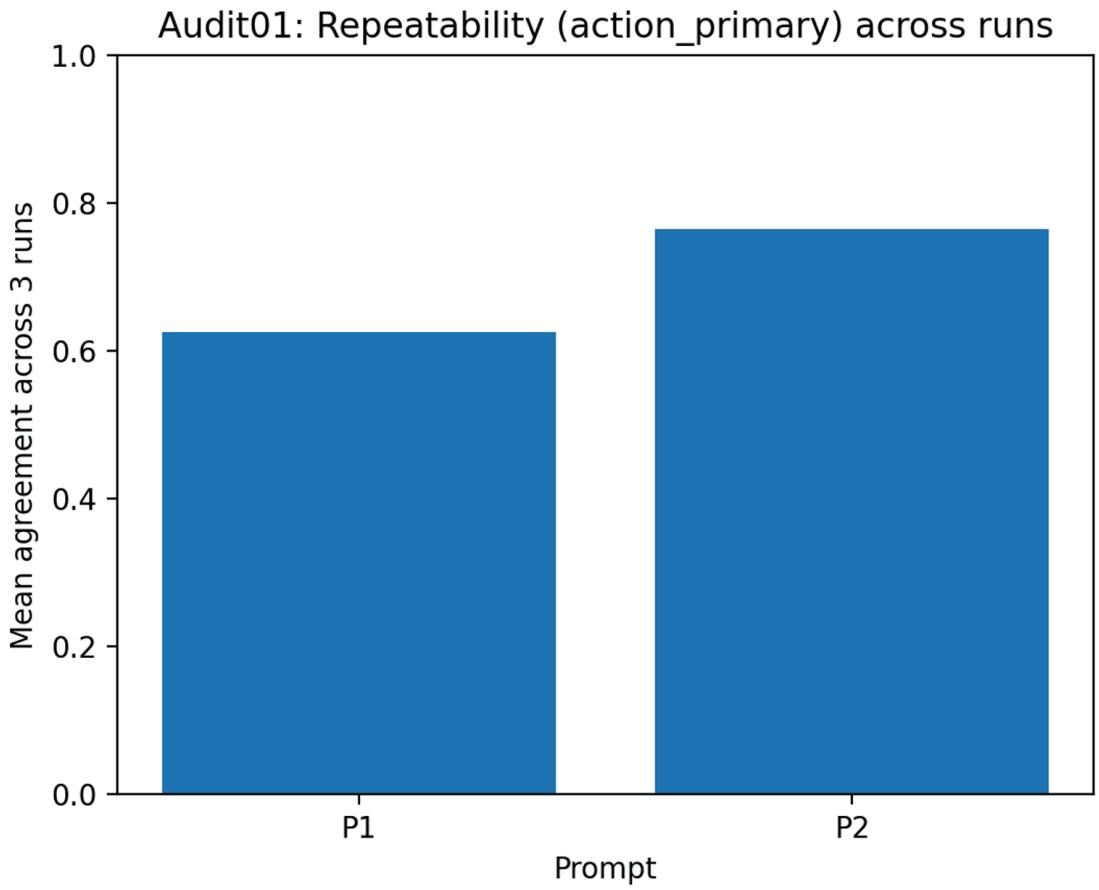

Figure 6. *Audit01 repeatability across runs (mean agreement for action_primary).*

5.7. Controlled follow-up: decoding-controlled reruns (temperature 0 vs default)

To disentangle prompt framing effects from stochastic decoding noise and to address the limited repeat counts in the initial experiments, we conducted a controlled follow-up rerun on audit01_base24 and redflag24 using per-item isolated execution and explicitly specified decoding parameters (Section 3.4). We compared a near-deterministic configuration (det_t0; temperature 0.0 with a fixed seed) to a stochastic configuration matching documented defaults (stoch_default; temperature 1.0, top-p 0.95 with per-run seeds). Across conditions we targeted 10 repeats; completed repeats ranged from 8 to 10 due to occasional transient failures.

Prompt sensitivity remained high under both decoding regimes. Across completed runs, the mean P1 vs P2 mismatch rate for action_primary was 78.7–82.1% depending on slice and decoding configuration, and the mean Jensen–Shannon divergence between P1 and P2 action distributions within a run was 0.148–

0.196 bits (Table 1). These results indicate that the large prompt effects observed in Study are not explained solely by stochastic sampling under default decoding.

Slice	Decoding configuration	Runs (P1/P2)	Mismatch (action_primary)	JSD(P1,P2) (bits)
audit01_base24 (non–red-flag)	det_t0 (temp=0)	10/10	82.1% [77.2–87.0]	0.148 [0.114–0.183]
audit01_base24 (non–red-flag)	stoch_default (temp=1.0, top-p=0.95)	9/9	81.9% [76.5–87.4]	0.162 [0.109–0.214]
redflag24 (red-flag)	det_t0 (temp=0)	10/9	78.7% [75.2–82.2]	0.183 [0.142–0.223]
redflag24 (red-flag)	stoch_default (temp=1.0, top-p=0.95)	9/8	80.7% [74.2–87.3]	0.196 [0.168–0.224]

Table 1. Controlled follow-up: prompt mismatch and distribution divergence (P1 vs P2) under explicit decoding configurations.

Decoding configuration, however, materially affected run-to-run stability. Using per-pair modal agreement as a repeatability measure, det_t0 increased mean agreement relative to stoch_default in three of four slice×prompt settings (Holm-adjusted paired Wilcoxon tests; Table 2). The exception was P2 in the red-flag slice, where det_t0 and stoch_default had similar repeatability.

Slice	Prompt	Modal agreement (det_t0)	Modal agreement (stoch_default)	p-value (Holm; one-sided)
audit01_base24 (non–red-flag)	P1	0.558	0.394	0.000584
audit01_base24 (non–red-flag)	P2	0.571	0.407	0.008
redflag24 (red-flag)	P1	0.529	0.366	0.000373
redflag24 (red-flag)	P2	0.551	0.510	0.251

Table 2. Controlled follow-up: repeatability (modal agreement for action_primary) under det_t0 vs stoch_default.

Notably, the average action distribution shift induced by changing decoding configuration was smaller than the prompt-induced shift. When aggregated across runs, det_t0 vs stoch_default produced Jensen–Shannon divergences of 0.018–0.073 bits for the marginal action_primary distribution, whereas P1 vs P2 divergences within a run were approximately 0.15–0.20 bits (Table 1). In this task, decoding parameters therefore appear to modulate variance and repeatability more strongly than the mean output distribution, while prompt framing shifts the mean distribution directly.

5.8. 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[18]. 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.

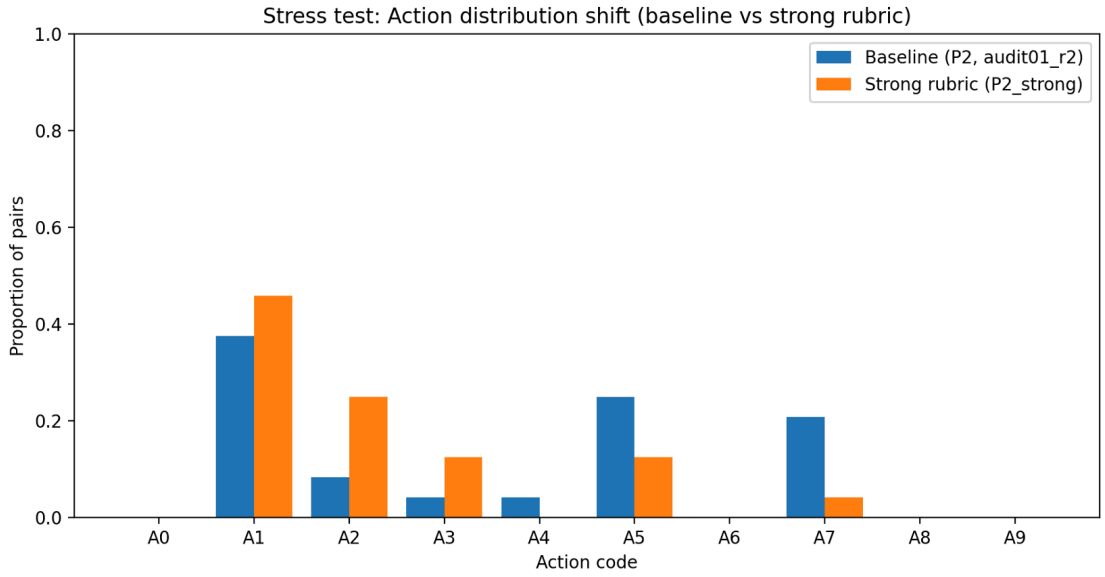

Figure 7. *Stress test action distribution shift (baseline vs strong rubric).*

5.9. 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[19][20].

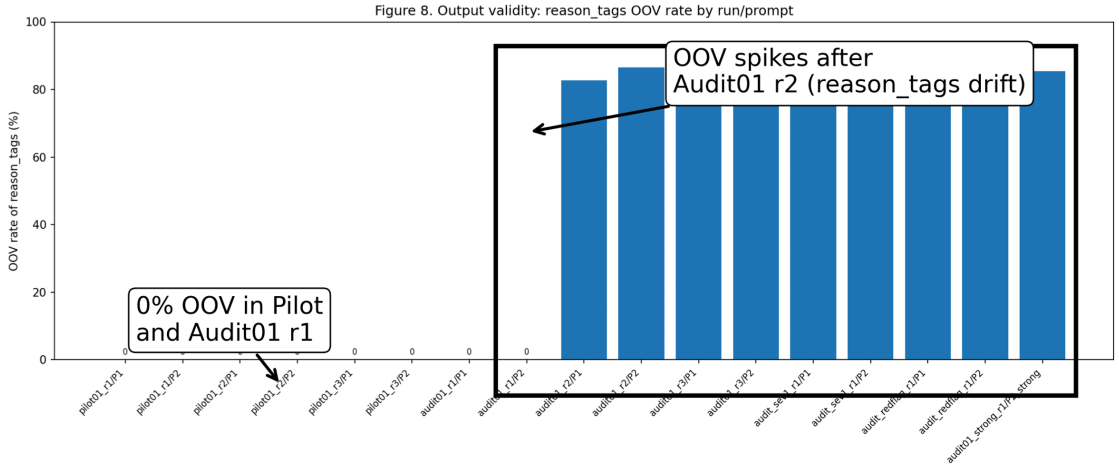

Figure 8. *Output validity: reason_tags OOV rate by run/prompt. Annotation highlights the step-change from 0% OOV (Pilot, Audit01 r1) to high OOV in subsequent audits.*

doi.org/10.32388/BEoZBC.2

5.10. 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
Controlled follow-up (8–10 repeats): prompt mismatch (action_primary)	mean mismatch 78.7–82.1% across slices and decoding; mean JSD(P1,P2) 0.148–0.196 bits
Controlled follow-up: decoding effect on repeatability	det_t0 increases mean modal agreement by +0.12–0.16 in 3/4 settings (Holm-adjusted p<0.01); red-flag P2 shows no clear gain

Table 3. Summary of key findings across experiments.

5.11. 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[21]. In the red-flag slice, the model most often

selected high-urgency actions (A8/A9) for these vignettes, especially under P2 (Table 4). 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.[21]	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.[22][23]	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.[24]	P1: A8; P2: A8

Table 4. Illustrative comparison between selected red-flag vignettes and public triage guidance.

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[25][26]. 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[27]. 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[28][29]. 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. In the initial slice audits, mismatch was low in a mild non–red-flag slice (severity=1) but higher in a red-flag slice. In the controlled follow-up, which focused on moderate-to-severe scenarios (audit01_base24 and redflag24), mismatch remained high in both slices under both deterministic and stochastic decoding (Table 1). Taken together, these results suggest that instability concentrates in boundary cases where multiple actions are defensible, and that "red-flag" status is one useful stratification but not the only driver; severity and vignette ambiguity also matter. For healthcare-adjacent simulations, this matters because boundary regions often correspond to high-stakes decision contexts, and instability there has disproportionate interpretive impact. Practically, reporting only an average mismatch rate over a mixed scenario set can mask important conditional failure modes, motivating slice-aware reporting as a default.

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[19][20][30]. 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[30].

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[17]. 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[17]. 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[16]. 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. Consistent with this guidance, our controlled follow-up shows that moving from default stochastic decoding to temperature 0 increases repeatability (modal agreement) in most settings (Table 2) without eliminating large prompt-induced distribution shifts (Table 1).

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 (e.g., public health behavior modeling), 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 5.11 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 initial Pilot and slice audits use 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. In response to reviewer feedback, we therefore add a controlled follow-up with 8–10 repeats per condition (Section 5.7), which produces less-quantized repeatability estimates and allows more stable comparisons across decoding configurations. 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., ≥10) remain 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. While the initial experiments did not lock down decoding parameters, we partially address this confound by adding a controlled decoding follow-up (Section 5.7) that explicitly contrasts temperature 0.0 versus default stochastic decoding under otherwise matched conditions. Nevertheless, we do not provide a full parameter sweep (temperature × topP × topK) nor do we pin server-side model revisions; broader sweeps across temperatures and model versions remain important to separate decoding-driven variance from prompt-driven variance more comprehensively. 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, mismatch rates, or apparent repeatability.

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 severely underestimated by within-batch designs, and decoding configuration can materially affect repeatability.

In a controlled follow-up with explicit parameter settings and 8–10 repeats, large prompt-induced shifts persisted even under near-deterministic decoding (temperature 0), while temperature primarily modulated run-to-run stability. We recommend multi-run, isolated-prompt evaluations with explicit slice audits, full reporting of model identifiers and decoding parameters, and (at minimum) reporting results at both a near-deterministic and a default stochastic operating point. Schema validation and, where appropriate, structured output enforcement should be treated as first-class experimental factors rather than assumed-neutral implementation details.

Statements and Declarations

Funding

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

Potential 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. ^*Aher GV, Arriaga RI, Kalai AT (2023). "Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies." ICML 2023 (PMLR 202). 337–371. https://proceedings.mlr.press/v202/aher23a.html.*
2. ^*Argyle LP, Busby EC, Fulda N, Gubler J, Rytting C, Wingate D (2023). "Out of One, Many: Using Language Models to Simulate Human Samples." Polit Anal. **31**(3):337–351. doi:10.1017/pan.2023.2.*
3. ^*Sun S, Lee E, Nan D, Zhao X, Lee W, Jansen BJ, Kim JH (2024). "Random Silicon Sampling: Simulating Human Sub-Population Opinion Using a Large Language Model Based on Group-Level Demographic Information." arXiv. https://arxiv.org/abs/2402.18144.*
4. ^*Zhou M, Yu L, Geng X, Luo L (2024). "ChatGPT vs Social Surveys: Probing the Objective and Subjective Human Society." arXiv. https://arxiv.org/abs/2409.02601.*
5. a, b*Tjuatja L, Chen V, Wu ST, Talwalkar A, Neubig G (2023). "Do LLMs Exhibit Human-Like Response Biases? A Case Study in Survey Design." arXiv. https://arxiv.org/abs/2311.04076.*
6. ^*Mukherjee S, Adilazuarda MF, Sitaram S, Bali K, Aji AF, Choudhury M (2024). "Cultural Conditioning or Placebo? On the Effectiveness of Socio-Demographic Prompting." arXiv. https://arxiv.org/abs/2406.11661.*
7. ^*Tseng YM, Huang YC, Hsiao TY, Chen WL, Huang CW, Meng Y, Chen YN (2024). "Two Tales of Persona in LLMs: A Survey of Role-Playing and Personalization." Findings EMNLP 2024. 16612–16631. doi:10.18653/v1/2024.findings-emnlp.969.*
8. ^*Park JS, O'Brien JC, Cai CJ, Morris MR, Liang P, Bernstein MS (2023). "Generative Agents: Interactive Simulacra of Human Behavior." UIST 2023. doi:10.1145/3586183.3606763.*
9. ^*Reichenpfader D, Denecke K (2024). "Simulating Diverse Patient Populations Using Patient Vignettes and Large Language Models." CL4Health @ LREC-COLING 2024. 20–25. https://aclanthology.org/2024.cl4health-1.3/.*
10. ^*Bender EM, Gebru T, McMillan-Major A, Shmitchell M (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" FAccT 2021. 610–623. doi:10.1145/3442188.3445922.*

11. ^Bisbee J (2024). "Synthetic Replacements for Human Survey Data? The Perils of Large Language Models." Polit Anal. **32**(4):401–416. doi:10.1017/pan.2024.5.

12. ^Dillion D, Tandon N, Gu H, Gray K (2023). "Can AI Language Models Replace Human Participants?" Trends Cogn Sci. **27**(11):597–600. doi:10.1016/j.tics.2023.08.001.

13. ^Wang A, Morgenstern J, Dickerson JP (2025). "Large Language Models That Replace Human Participants C an Harmfully Misportray and Flatten Identity Groups." Nat Mach Intell. **7**(3):400–411. doi:10.1038/s42256-0 25-00986-z.

14. ^Santurkar S, Durmus E, Ladhak F, Lee C, Liang P, Hashimoto T (2023). "Whose Opinions Do Language Mo dels Reflect?" ICML 2023 (PMLR 202). 29971–30004. https://proceedings.mlr.press/v202/santurkar23a.html.

15. ^Zhu K, Zhao Q, Chen H, Wang J, Xie X (2024). "PromptBench: A Unified Library for Evaluation of Large Lan guage Models." J Mach Learn Res. **25**(254):1–22. https://jmlr.org/papers/v25/24-0034.html.

16. a, bGoogle Cloud (2025). "Gemini 2.5 Flash (Vertex AI Model Documentation)." Google Cloud. https://docs.cl oud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash.

17. a, b, cGoogle Cloud (2026). "Content Generation Parameters | Generative AI on Vertex AI." Google Cloud. htt ps://docs.cloud.google.com/vertex-ai/generative-ai/docs/multimodal/content-generation-parameters.

18. ^Lin J (1991). "Divergence Measures Based on the Shannon Entropy." IEEE Trans Inf Theory. **37**(1):145–151. d oi:10.1109/18.61115.

19. a, bGoogle AI for Developers (2026). "Models (Gemini API Documentation)." Google AI for Developers. http s://ai.google.dev/gemini-api/docs/models.

20. a, bGoogle AI for Developers (2026). "Structured Outputs (Gemini API Documentation)." Google AI for Devel opers. https://ai.google.dev/gemini-api/docs/structured-output.

21. a, bUniversity Hospitals Sussex NHS Foundation Trust (2025). "Headaches: Emergency Department Patient Leaflet." University Hospitals Sussex NHS Foundation Trust. https://www.uhsussex.nhs.uk/resources/heada ches-emergency-department-patient-leaflet/.

22. ^NHS (2026). "Chest Pain." NHS. https://www.nhs.uk/symptoms/chest-pain/.

23. ^NHS (2026). "Shortness of Breath." NHS. https://www.nhs.uk/symptoms/shortness-of-breath/.

24. ^NHS (2026). "Anaphylaxis." NHS. https://www.nhs.uk/conditions/anaphylaxis/.

25. ^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, Ch en 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." Adv Neural Inf Process Syst. https://arxiv.org/abs/2005.14165.

26. ^*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?" Proc EMNLP 2022. https://aclanthology.org/2022.emnlp-main.759/.*
27. ^*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.*
28. ^*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.*
29. ^*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.*
30. 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.*

Supplementary data: available at https://doi.org/10.32388/BE0ZBC.2

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.