IDIAP RESEARCH REPORT

DIALFORK: GOING BEYOND FACTUALITY TO EVALUATE PROCEDURAL HALLUCINATION IN CONVERSATIONAL AGENTS

Sergio Burdisso Esaú Villatoro-Tello[a]
Dairazalia Sanchez-Cortes Yanis Labrak
Jordi Luque Milos Cernak Petr Motlicek

Idiap-RR-05-2026

OCTOBER 2026

[a]Idiap

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

🔱 DIALFORK: Going Beyond Factuality to Evaluate Procedural Hallucination in Conversational Agents

Sergio Burdisso[1] **Esaú Villatoro-Tello**[1] **Dairazalia Sánchez-Cortés**[1]
Yanis Labrak[1] **Jordi Luque**[3] **Milos Cernak**[2] **Petr Motlicek**[1,4]
[1]Idiap Research Institute, Martigny, Switzerland
[2]Logitech, Lausanne, Switzerland
[3]Telefónica Innovación Digital, Barcelona, Spain
[4]Brno University of Technology, Brno, Czech Republic
sergio.burdisso@idiap.ch

Abstract

Large language models (LLMs) deployed as conversational agents are increasingly expected to follow procedural logic. While factual hallucination has received extensive attention, a different failure mode has been overlooked: *procedural hallucination*, in which an agent produces individually fluent and factually plausible utterances while deviating from the intended conversational process. We introduce DIALFORK, the first benchmark to measure this failure mode on raw dialogue text. DIALFORK is built by *forking* conformant dialogues at controlled turn positions and injecting one of four hallucination types, yielding 36,278 dialogues across 24 tasks in 12 domains; 30 human annotators validate the construction. We define three evaluation tasks that characterize procedural deviation at three granularities: how far a set of dialogues deviates from conformant ones (monitoring deployed agents), how far a single dialogue does (procedural retrieval), and at which turn a dialogue first goes off-track (real-time guardrails). For the first two, we propose dialog-flow-aware metrics, FLOWKL and FLOWSCORE, that compare dialogues by *how they unfold* rather than their *content*. At the dialogue level, FLOWSCORE significantly outperforms all baselines across seven encoders, showing that semantic similarity is no substitute for procedural similarity. We release DIALFORK under open licenses at github.com/idiap/dialfork, hoping the community will measure conversational *process* alongside content.

1 Introduction

As LLMs are deployed in increasingly agentic roles (customer support, scheduling, guided troubleshooting, clinical intake), they are implicitly expected to follow structured procedural logic: a sequence of prescribed steps that moves a conversation from a user's initial request to a completed goal (Guo et al., 2024; Alismail and Lanquillon, 2025). This expectation extends well beyond classical task-oriented dialogues (TOD); it applies to any system where the *process* of the conversation matters as much as the content of individual utterances.

The dominant conception of hallucination in LLMs concerns *factual* errors (Maynez et al., 2020; Ji et al., 2023; Huang et al., 2025), for instance: confident assertions of incorrect dates, invented product details, or wrong numbers. A qualitatively different and practically critical failure mode has, however, received comparatively little attention: an agent that produces entirely truthful, fluent utterances while violating the procedural logic it was instructed to follow. The closest prior notions, hallucination in inter-agent communication (Lin et al., 2025) and faithfulness to grounding knowledge in dialogue (Dziri et al., 2022), concern the accuracy of *content*, not adherence to a conversational *process*.

We call this **procedural hallucination**: a breakdown in *process* rather than *content*. A procedurally hallucinating agent may skip required steps, pursue a different goal than instructed, repeat already-completed actions, or introduce conflicting objectives that derail task completion, all while generating individually plausible responses. Crucially, this failure is *orthogonal* to factual correctness: no single utterance need be false for the overall conversation to be wrong. Standard hallucination benchmarks (Li et al., 2023; Lin et al., 2022) are blind to such failures because they evaluate the truthfulness of content, not the correctness of the conversational process.

To address this gap, we present DIALFORK, a benchmark constructed by *forking* dialogues at controlled turn positions (Figure 1) to simulate different types of procedural hallucinations. Our main contributions are the following: (a) We define *procedural hallucination* as a phenomenon dis-

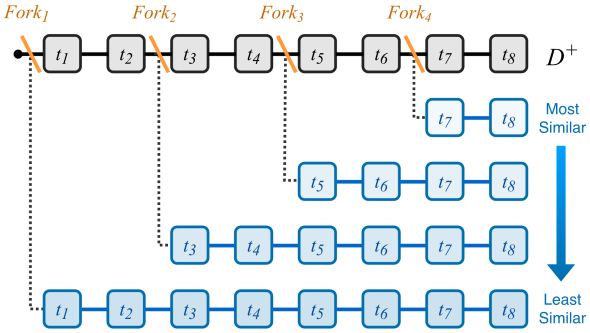


Figure 1: Construction of DIALFORK by controlled forking. To produce a forked variant of a conformant dialogue D^+, the simulation is re-run with the same random seed (reproducing the conformant prefix), and a hallucination instruction is injected into the system agent's context just before the fork turn, causing the continuation to diverge. Earlier forks share less prefix and yield more deviant dialogues. This induces a natural ground-truth ordering by *procedural* (dialog-flow) similarity, distinct from pure semantic similarity.

tinct from factual hallucination, and decompose it into four complementary types grounded in a general workflow model (§3). (b) We release DIALFORK,[1] a 36,278-dialogue benchmark constructed via controlled forking that spans 24 tasks across 12 domains and four temporal deviation depths, along with its seed-controlled multi-agent generation pipeline, applicable to any workflow-annotated dialogue dataset, and validate its construction in a human study with 30 annotators (§4). (c) We define three evaluation tasks that characterize procedural deviation at three granularities, directly from raw dialogue text: how far a set of dialogues deviates from conformant ones, how far a single dialogue does, and at which turn a dialogue first goes off-track (§5). (d) Enabled by the new dataset, we introduce FLOWKL and FLOWSCORE, two dialog-flow-aware distance metrics that compare dialogues by how they unfold rather than by their content, and report initial baselines on all three tasks as a starting point for future work (§6).

2 Related Work

Hallucination in LLMs. The dominant view concerns factual accuracy of generated content (Maynez et al., 2020; Kryściński et al., 2020; Lin et al., 2022; Mallen et al., 2023). Surveys (Ji et al., 2023; Zhang et al., 2025b; Huang et al., 2025) taxonomize hallucination along faithfulness and factuality dimensions, and dialogue-level benchmarks such as DiaHalu (Chen et al., 2024b) extend them to multi-turn conversations; all these categories, however, concern *what* is said, not the correctness of *how* a conversation unfolds.

Procedural adherence and dialogue similarity. A parallel line studies whether systems follow prescribed steps, from policy-constrained action sequences (Chen et al., 2021) and procedure-guided action prediction (White et al., 2023) to condition–action instruction following (Ghazarian et al., 2025) and workflow generation (Qiao et al., 2025), typically over annotated actions, intents, or slots. Dialogue similarity is largely text similarity: embeddings collapse the turns into one vector (Reimers and Gurevych, 2019; Zhou et al., 2022) compared by cosine, discarding turn order; TaskDiff (Bhaumik et al., 2023) relies on annotated intents and slots. These benchmark how well models follow procedural instructions; DIALFORK instead benchmarks methods that measure how far a conversation *deviates* from a procedure, *directly on raw dialogue text*, whether an LLM or a human produced it.

3 Procedural Hallucination

3.1 Formal Model

Most conversational systems do not declare their procedural logic explicitly: states, available actions, and transitions are implicit in the agent's instructions, the underlying business process, or the assumed conversational norms. To talk about *deviations* from this logic precisely, we introduce a lightweight abstraction that does not assume any particular implementation. We model the intended logic of a conversational agent as a tuple $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{O})$, where $\mathcal{S}$ is a set of conversational states (*e.g.,* steps or phases of the interaction), $\mathcal{A}$ is a set of agent actions (*e.g.,* the communicative functions it can perform), $\delta : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ is a transition function from the current state and user input $\mathcal{U}$ to the next state, and $\mathcal{O}$ is the agent's objective.

This formulation is intentionally general. It applies to any goal-directed conversational agent. For example, a task-oriented dialogue system has the dialogue policy as δ and task completion (*e.g.,* booking a hotel room, retrieving a balance) as $\mathcal{O}$; a medical-intake assistant has a clinical protocol as δ and patient triage as $\mathcal{O}$; a customer-support agent has a troubleshooting procedure as δ and issue resolution as $\mathcal{O}$. We define a *conformant* dialogue

[1] https://github.com/idiap/dialfork: dataset, generation pipeline, evaluation code, and annotation tool.

Table 1: Procedural hallucination types and the components of $\mathcal{W} = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{O})$ they corrupt (✓: preserved; ×: replaced; ∼: partially overridden). Simplified example (a real one is in Appendix G): after the shared opening, each row shows only the first two turns from the fork turn t_k on; every version then continues (...) as a full dialogue. Gray: intended workflow; blue: corrupted.

Type	$(\mathcal{S}, \mathcal{A})$	δ	$\mathcal{O}$	**What is broken**	**Example** (simplified): first two turns after the fork
Shared opening before the fork					**User:** I'd like to book a viewing. **System:** Your name? **User:** Mark.
Conformant	✓	✓	✓	— (original)	**System:** Which apartment do you want to view? **User:** Center Apts. ...
WIC	×	×	✓	Workflow identity	**System:** Let's search apartments! budget? **User:** No, just a viewing! ...
SSPC	✓	×	✓	Step selection	**System:** Your viewing is booked! **User:** What?! Which one?! ...
GR	✓	✓	×	Task goal	**System:** Want to try co-living instead? **User:** No, no, I already know ...
CSC	✓	✓	∼	Goal conflict	**System:** Just to confirm, your name is Mark? **User:** Yes, Mark. ...

D^+ as one whose turns are produced in accordance with $\mathcal{W}$.

3.2 Hallucination Types

We define four complementary types, each targeting a distinct component of $\mathcal{W}$ (summarized in Table 1, with an example of each):

• **Workflow Identity Corruption (WIC).** $\mathcal{W}$ is replaced by a different workflow $\mathcal{W}'$; the agent follows a valid workflow but the wrong one (modeling task disambiguation failures and prompt injection).

• **Step Selection Policy Corruption (SSPC).** $\mathcal{O}$ and $(\mathcal{S}, \mathcal{A})$ are preserved but δ is replaced by a corrupted policy δ' that, *e.g.,* repeats, skips, loops, or randomizes steps.

• **Goal Replacement (GR).** $\mathcal{W}$ is preserved but the objective $\mathcal{O}$ is replaced by a qualitatively different goal $\mathcal{O}'$ (*e.g.,* collecting feedback instead of completing a booking).

• **Conflicting System Constraints (CSC).** $\mathcal{O}$ is retained but augmented with a competing auxiliary objective $\mathcal{O}_c$ that intermittently overrides $\mathcal{O}$ (*e.g.,* always responding with a question, or confirming every detail).

In practice, for both diversity and fine-grained controllability, we define a small pool of concrete variants per hallucination type (five corrupted policies for SSPC such as *repeat*, *skip*, and *invert order*, six alternative goals for GR, and six conflicting constraints for CSC); the full lists are in Appendix B.

4 The DIALFORK Dataset

4.1 Construction Pipeline

STAR as a generation basis. We base DIALFORK on STAR (Mosig et al., 2020), which spans 24 tasks across 12 domains (hotel, ride, restaurant, medical, party, spaceship, trivia, weather, etc.). STAR is uniquely suited to our pipeline because its dialogues were originally produced by *human role-play*: for each dialogue, one human participant was given the system role, with the task's workflow graph and per-node response templates (Appendix C), and another was given the user role, with a persona, a task, and a goal; the two then conversed in these roles. Crucially, STAR releases not only the resulting dialogues but also these role-play instructions, exactly as they were given to the two human participants of each dialogue (see Appendix D for a complete example). We give the same instructions to two LLM agents that take the place of the humans: the system agent receives the instructions given to the human who played the system, and the user agent those given to the human who played the user. To our knowledge, no other public dialogue corpus releases both an explicit workflow for every task, which serves as procedural ground truth, and the instructions given to the humans who produced each conversation. Since every DIALFORK dialogue is newly generated, only STAR's instructions and workflows, not its original dialogues, enter the benchmark.

Conformant dialogue generation. For each STAR scenario, we use SDIALOG (Burdisso et al., 2026) to run the two agents: the **system agent** is instructed to follow the workflow faithfully, and the **user agent** to interact naturally without volunteering all details upfront. Both agents are `Gemma4-31B` (Google DeepMind, 2026) served locally[2]; prompts and full implementation details are in Appendix E. The two agents converse until the task is completed or a per-task turn budget is reached. After filtering for dialogues that include enough system turns to admit all four fork positions, we retain an initial set of **2,134 valid conformant**

[2]Selected among several open-weight LLMs from pilot dialogues (Appendix E). An open-weight model stays fixed (closed models may change behind the same name), lets anyone regenerate or extend DIALFORK, and avoids the cost of a commercial API at this scale.

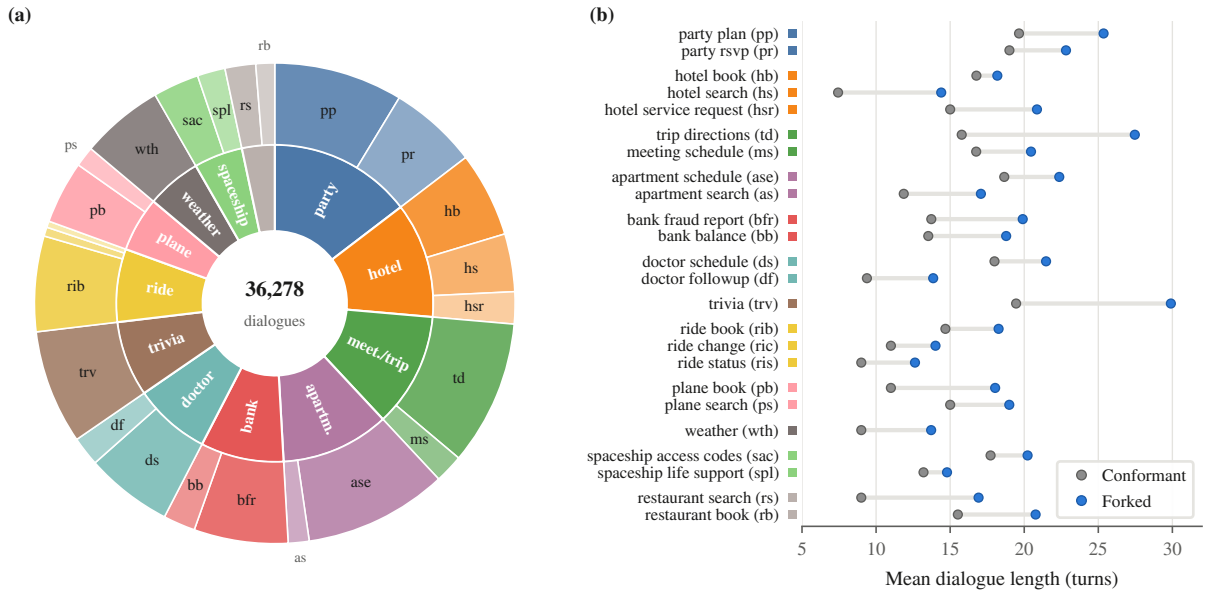


Figure 2: Per-task composition of DIALFORK. (a) Number of dialogues per STAR domain (inner ring) and task (outer ring; abbreviations as in (b)). Since every conformant dialogue yields 16 forks, the proportions are the same for conformant and forked dialogues. (b) Mean dialogue length per task, conformant vs. forked, grouped by domain in the same order and colors as (a).

dialogues.

Forked hallucination generation via seeded replay. For each conformant dialogue D^+ with T system turns, we identify four *fork points* at evenly-spaced positions $t_k = 1 + \text{round}((k-1) \cdot T/4)$, $k \in \{1, 2, 3, 4\}$. To produce a forked variant, we re-run the same scenario with the same random seed, so that turns 1 through $t_k - 1$ are again generated under the intended workflow and reproduce those of D^+;[3] before turn t_k is generated, a hallucination instruction is added as a system message to the system agent's context, specifying the type and the sampled variant (drawn from the per-type pool defined in §3; full prompts in Appendix F). From t_k onward, the system agent follows the injected instruction and produces turns that diverge from D^+ in the manner specified by the chosen hallucination type (Table 1; worked examples in Appendix G). Earlier forks produce more deeply deviant dialogues, while later forks produce subtler deviations, naturally inducing a similarity ranking over the forked dialogues (Figure 1).

Quality assurance. DIALFORK's usefulness rests on a single property: *forking must produce controlled deviation from the conformant baseline*. As Task 1 will show (§6), automatic metrics confirm this property across the whole dataset by recovering the expected fork ordering, but they cannot tell *how* or *when* each fork deviates. We therefore ran a human study: 30 annotators judged 428 conversations from five tasks (526 judgments), answering in stages so that they rated each conversation before learning how it was built. The study confirms the four labels the benchmark relies on. (i) *Fork order:* for the types that break the workflow (WIC, SSPC, GR), blind conformance ratings rise with fork position, from 14/100 at k=1 to 58 at k=4 and 94 for conformant dialogues (τ_b=0.64 across the four forks of a dialogue; Figure 3a). (ii) *Hallucination type:* told which instruction had been injected, annotators confirmed it in 91% of forks, vs. 8% for foil instructions on conformant dialogues. (iii) *Fork turn:* they placed the onset of the injected behavior within one turn of t_k in 77% of cases (exactly at t_k in 67%; Figure 3b). (iv) *Prefix:* the turns before the fork follow the workflow in 95% of cases. Appendix N details the study design and further results, including inter-annotator agreement; we release the annotation tool and the anonymized judgments along with DIALFORK.

[3] Up to minor rewordings that the LLM serving stack may introduce despite the fixed seed. These do not affect the construction: every turn before t_k is generated under the intended workflow, and every turn from t_k on under the corrupted one.

4.2 Dataset Statistics

Each of the 2,134 conformant dialogues yields 16 forked variants (4 hallucination types × 4 fork positions), for a total of **36,278 dialogues** (about

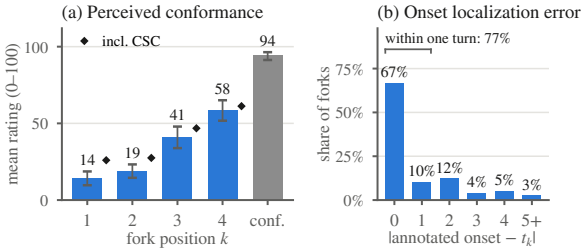


Figure 3: Human validation of DIALFORK (Appendix N). (a) Mean blind conformance rating per fork position k for WIC, SSPC, and GR forks (bars, with 95% bootstrap CIs) and for all four types (markers), and for conformant dialogues. (b) Distance, in assistant turns, between the annotated onset of the injected behavior and the fork turn t_k.

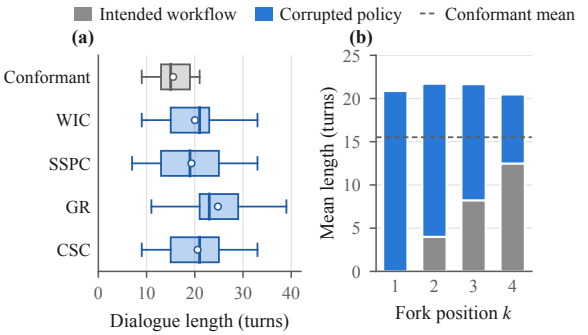


Figure 4: DIALFORK statistics. (a) Dialogue length per split (box: interquartile range; whiskers: 5th–95th percentiles; line: median; dot: mean). (b) Mean number of turns per fork position k, split into turns generated before the fork, under the intended workflow, and after it, under the corrupted policy (averaged over the four hallucination types).

750K turns and 10.4M words, generated in about 33 GPU-days), 8,536 per hallucination type (per-task composition in Figure 2; exact counts in Tables 5 and 6, Appendix A). Figure 4a shows that forked dialogues are longer than conformant ones (21.18 vs. 15.52 turns on average), with GR the longest (verbose goals such as information collection add content) and SSPC the shortest (skipped or repeated steps add little content). As k increases (Figure 4b), the intended-workflow prefix grows from 0 to 12.5 turns while the corrupted continuation shrinks from 20.9 to 8.0, keeping total length roughly constant.

5 Evaluation Tasks and Methods

We define three evaluation tasks on DIALFORK at increasing levels of granularity: set-level, dialogue-level, and turn-level. The first two probe procedural (dialog-flow) deviation as a measurable distance,

Table 2: Task 1 (Set-to-Set Distance): Kendall's τ between predicted and ground-truth fork-position rankings, averaged across the 24 tasks. Top block: controls; **bold**: best per column; shading: darker is better. All three distance metrics are significantly above chance ($p < 0.001$); differences among them are not significant ($p \geq 0.43$; Appendix J).

Method	WIC	SSPC	GR	CSC	Avg.
Random	0.000	0.000	0.000	0.000	0.000
Turn-count diff.	−0.076	−0.340	0.278	−0.208	−0.087
Word-count diff.	0.222	0.208	0.583	−0.208	0.201
FBD	**0.972**	0.792	**0.986**	**0.903**	**0.913**
PRD	0.931	0.764	**0.986**	0.833	0.878
FLOWKL	0.847	**0.903**	0.944	0.833	0.882

an axis distinct from semantic similarity: two dialogues can be near-identical semantically (*e.g.*, same topics, same vocabulary) yet follow entirely different conversational procedures, and vice versa. The third localizes the deviation, asking where in a dialogue a hallucination first manifests. Throughout, dialogues are raw transcripts, with no annotations of any kind and no access to the model that produced them: the tasks measure procedural deviation in the text itself, whatever its source (an LLM agent or a human operator), rather than how prone a particular model is to deviate.

5.1 Task 1: Set-to-Set Distance Ranking

Consider a company that deploys an LLM agent to handle its routine customer requests, or that upgrades its assistant to a newer model. Before deployment, it needs to know whether the new agent's conversations unfold like the reference ones (*e.g.*, past conversations handled by its human agents or by the previous assistant), and, after deployment, whether they drift over time. Neither question concerns single dialogues; Task 1 captures this setting. *Input:* two sets of dialogues, a reference and a target; *output:* a scalar distance between them. To evaluate it with DIALFORK, for each hallucination type h and each of the 24 tasks of DIALFORK, we group the forked dialogues by their fork position to obtain four sets $\mathcal{D}_k^h$ ($k \in \{1, 2, 3, 4\}$), and take the set of conformant dialogues for that task, $\mathcal{D}^+$, as the reference. Writing $\mu_k \equiv \mu(\mathcal{D}^+, \mathcal{D}_k^h)$ for the distance under set-distance metric μ, we expect the natural *procedural-similarity* ordering $\mu_1 > \mu_2 > \mu_3 > \mu_4$, since earlier forks share less of the conformant prefix and are therefore more procedurally deviant. We report Kendall's τ (Kendall,

1938) between the metric-induced ranking and this ground-truth ordering, averaged across the 24 tasks. Because this ordering follows directly from how forks are constructed, Task 1 also serves as a *sanity check* of DIALFORK itself: if independent metrics recover it, the forked dialogues indeed depart from the conformant ones at their designated positions.

Baselines. To the best of our knowledge, FBD and PRD, introduced together by Xiang et al. (2021), are the only two existing distribution-wise metrics specifically designed to compare sets of dialogues. Both operate on embeddings of consecutive (system, user) turn pairs extracted by a pretrained language model. **FBD** (Fréchet BERT Distance) fits a multivariate Gaussian to each set's embedding distribution and reports the Fréchet distance between the two Gaussians. **PRD** (Precision–Recall Distance) treats the two sets as distributions and reports the maximum harmonic mean of precision and recall over a sweep of relative-density thresholds (Sajjadi et al., 2018).

Controls. Since forks are longer than conformant dialogues (Figure 4), **Turn-/Word-count diff.** use as distance $|\overline{\text{len}}(\mathcal{D}_k^h) - \overline{\text{len}}(\mathcal{D}^+)|$, the difference in mean length (in turns or words), checking whether length alone recovers the ordering. **Random** is the exact chance level.

FLOWKL. We propose FLOWKL, an interpretable alternative grounded in an explicit dialog flow graph G, computed in three steps:

1. **Graph construction.** Following Burdisso et al. (2024), we cluster all utterances in $\mathcal{D}^+$ in the embedding space of Dialog2Flow (D2F), an encoder that groups utterances by communicative function, to obtain workflow-step nodes; directed edges encode observed transitions with frequency weights, yielding a bigram model $P_G(n_t \mid n_{t-1})$ over node sequences (Figure 6 in Appendix shows the actual graph induced for the *doctor_schedule* task).
2. **Per-dialogue flow likelihood.** Each turn u_t of a dialogue $D = (u_1, \ldots, u_T)$ is mapped to the node n_t whose centroid is nearest to its D2F embedding $\phi(u_t)$. We then score how well D follows the reference workflow as the geometric mean of its transition probabilities under G:

$$\mathcal{L}(D) = \Big(\prod_{t \in \mathcal{K}_D} \hat{P}_G(n_t \mid n_{t-1}) \Big)^{1/|\mathcal{K}_D|},$$

   where $\mathcal{K}_D$ is the set of transitions of D observed in G and $\hat{P}_G$ the Laplace-smoothed transition probability.
3. **Distribution distance.** Let $p_{\mathcal{D}}$ be the kernel density estimate of $\{\mathcal{L}(D) : D \in \mathcal{D}\}$ for a set $\mathcal{D}$. FLOWKL is finally the KL divergence between the reference and target sets:

$$\text{FLOWKL}(\mathcal{D}_{\text{ref}}, \mathcal{D}_{\text{tgt}}) = \text{KL}\big(p_{\mathcal{D}_{\text{ref}}} \,\|\, p_{\mathcal{D}_{\text{tgt}}}\big),$$

   with $\mathcal{D}_{\text{ref}} = \mathcal{D}^+$ and $\mathcal{D}_{\text{tgt}} = \mathcal{D}_k^h$.

Unlike FBD and PRD, which compare opaque embedding distributions, FLOWKL relies on an explicit, human-readable model of what conformant dialogues look like, the graph G, and measures how differently the two sets fit it, *i.e.,* how likely their dialogues are to follow its transitions.

5.2 Task 2: Dialog-to-Dialog Distance Ranking

Task 2 probes procedural (dialog-flow) similarity at the level of individual dialogues, as needed, for instance, to retrieve past conversations that unfolded like a given one (*e.g.,* to find how similar cases were handled). *Input:* two dialogues; *output:* a scalar distance between them. To evaluate it with DIALFORK, for each conformant dialogue D^+, the four forked dialogues D_k^h provide an unambiguous 4-item ranking of distance from D^+ (Figure 1), and a good dialogue-distance metric should recover this ranking. We report Kendall's τ between predicted and ground-truth rankings, averaged across all conformant dialogues. As, to our knowledge, no metric has been designed for dialogue-to-dialogue distance on raw dialogue text, we compare standard pooling baselines with two new flow-aware methods. All methods below define a distance between two dialogues D and D' (in our evaluation, $D = D^+$ and $D' = D_k^h$), with per-turn embedding sequences $E = (e_1, \ldots, e_n)$ and $E' = (e'_1, \ldots, e'_m)$.

Pooling baselines. **Mean-turn**: $\text{Mean}(D, D') = 1 - \cos(\bar{e}, \bar{e}')$, where $\bar{e}$ and $\bar{e}'$ average the per-turn embeddings. **Full-dialog**: $\text{Full}(D, D') = 1 - \cos(\phi(D), \phi(D'))$, where ϕ encodes the concatenated transcript once.

Controls. Turn-/Word-count diff. use $|\text{len}(D) - \text{len}(D')|$ (in turns or words) as distance; **Random** has expected $\tau = 0$. **Shared prefix** uses the number of opening turns D and D' share verbatim. This control exploits the construction mechanism itself and has no counterpart in real dialogues, which do

Table 3: Task 2 (Dialog-to-Dialog Distance): Kendall's τ between predicted and ground-truth fork rankings, averaged across all conformant dialogues. Top block: controls. **Bold**: best per encoder; **bold+underline**: best overall; shading: darker is better. Mean/Full = mean-turn/full-dialog embedding. FLOWSCORE's improvements over every other method are statistically significant for every encoder (one-sided Wilcoxon, Holm-corrected, $p < 0.001$; Table 7).

Encoder	Method	WIC	SSPC	GR	CSC	Avg.
Random		0.000	0.000	0.000	0.000	0.000
Turn-count diff.		0.025	0.251	0.226	0.002	0.126
Word-count diff.		0.220	0.242	0.390	0.040	0.223
Shared prefix		0.688	0.668	0.712	0.662	0.682
LaBSE	Mean	0.816	0.745	0.895	0.579	0.759
	Full	0.755	0.705	0.813	0.428	0.675
	FLOWDTW	0.842	0.738	0.907	0.479	0.742
	FLOWSCORE	**0.925**	**0.813**	**0.941**	**0.680**	**0.840**
Qwen3-Emb.	Mean	0.788	0.631	0.772	0.400	0.648
	Full	0.580	0.576	0.650	0.270	0.519
	FLOWDTW	0.844	0.720	0.881	0.440	0.721
	FLOWSCORE	**0.899**	**0.789**	**0.917**	**0.642**	**0.812**
SBERT	Mean	0.810	0.714	0.858	0.530	0.728
	Full	0.671	0.628	0.739	0.453	0.623
	FLOWDTW	0.843	0.727	0.892	0.492	0.739
	FLOWSCORE	**0.922**	**0.815**	**0.934**	**0.700**	**0.843**
BGE-M3	Mean	0.822	0.737	0.882	0.550	0.748
	Full	0.780	0.704	0.856	0.506	0.712
	FLOWDTW	0.823	0.727	0.891	0.468	0.727
	FLOWSCORE	**0.924**	**0.814**	**0.942**	**0.700**	**0.845**
D2F	Mean	0.848	0.743	0.895	0.561	0.762
	Full	0.817	0.724	0.843	0.603	0.747
	FLOWDTW	0.864	0.702	0.913	0.489	0.742
	FLOWSCORE	**0.925**	**0.798**	**0.936**	**0.675**	**0.834**
DSE	Mean	0.832	0.719	0.883	0.531	0.741
	Full	0.768	0.724	0.801	0.524	0.704
	FLOWDTW	0.848	0.698	0.897	0.505	0.737
	FLOWSCORE	**0.922**	**0.794**	**0.929**	**0.685**	**0.833**
Emb.Gemma	Mean	0.817	0.726	0.876	0.539	0.740
	Full	0.725	0.670	0.845	0.462	0.675
	FLOWDTW	0.833	0.701	0.886	0.463	0.721
	FLOWSCORE	**0.911**	**0.801**	**0.935**	**0.692**	**0.835**

not come with a conformant twin; we report it only to verify that our metrics do not reduce to it.

FLOWDTW. We align E and E' using Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978). Local costs are $c(i,j) = 1 - e_i^\top e'_j$ and the DTW recurrence $\Gamma(i,j) = c(i,j) + \min\{\Gamma(i-1,j), \Gamma(i,j-1), \Gamma(i-1,j-1)\}$ with $\Gamma(0,0) = 0$ gives the optimal aligned cost, normalized by the length L of the optimal warping path: $\text{FLOWDTW}(D, D') = \Gamma(n,m)/L$.

FLOWSCORE. A BERTScore-inspired (Zhang et al., 2020) matching applied at the turn level. We build $S(i,j) = e_i^\top e'_j$ and compute

$$P = \frac{1}{n}\sum_i \max_j S(i,j),\ R = \frac{1}{m}\sum_j \max_i S(i,j),$$

$F_{\text{flow}} = 2PR/(P+R)$, and define the distance $\text{FLOWSCORE}(D, D') = 1 - F_{\text{flow}}$. FLOWSCORE captures soft turn-level alignment without requiring the strict monotonic warping of DTW. Like BERTScore, FLOWSCORE is meant to operate on contextualized embeddings, which encode each element in its context and thus make the matching sensitive to order. To our knowledge, however, no contextualized *turn* encoders exist: with the sentence encoders used here, each turn is embedded independently, so our instantiation of FLOWSCORE does not capture explicit turn order. Thus, contextualized turn encoders are a natural direction for future work.

5.3 Task 3: Hallucination Turn Prediction

Task 3 probes whether a method can localize *where* a dialogue first goes off-track, as needed, for instance, by a guardrail that monitors a live conversation and intervenes as soon as the agent deviates from its intended process. The output is always the first hallucinated system turn, the hallucination *onset*, or none if the dialogue is conformant; for forked dialogues, the goal is to identify the fork point t_k as closely as possible. We consider three regimes of decreasing realism:

1. **Streaming**: the dialogue's turns arrive one at a time, and the method must decide after each turn whether the conversation has gone off-track, with no access to future turns. Its only other input is a set of conformant reference dialogues (*e.g.,* past human conversations), with no workflow. The most realistic and hardest setting.
2. **Non-streaming**: the complete dialogue, with no reference dialogues and no workflow.
3. **Non-streaming + workflow**: the complete dialogue *and* its ground-truth workflow, typically unavailable in practice; reported as a reference with privileged access to the workflow.

Streaming. *First*, for each dialogue we map turn embeddings to discrete nodes via the encoder's clustering (as in step 1 of FLOWKL, §5.1) and fit an N-gram language model over the conformant node sequences. Let $p_t = P_{\text{LM}}(n_t \mid n_{t-N+1}, \ldots, n_{t-1})$. Because the node assignment may be uncertain when the turn embedding $\phi(u_t)$ lies between clusters, we also consider weighting each transition by a *node confidence* (NC),

$$c_t = P_{\text{NC}}(n_t \mid u_t) = \frac{\exp(\sigma_1)}{\sum_{j=1}^{K}\exp(\sigma_j)},$$

Table 4: Task 3 (Turn Prediction): main results. MAE and conformant coverage (Cov.): lower is better. Hallucinated coverage and Score: higher is better. **Bold**: best within a regime; **bold+underline**: best overall. Shading: darker is better; baselines are in gray.

	Hallucination MAE ↓				Hallucination		Conformance		
Method	**CSC**	**GR**	**SSPC**	**WIC**	**Avg. MAE ↓**	**Cov. ↑**	**MAE ↓**	**Cov. ↓**	**Score ↑**
Baselines									
Random	6.78	8.56	6.28	7.08	7.18	100.00	8.16	100.00	0.94
First turn	2.83	2.83	2.83	2.83	2.83	100.00	14.17	100.00	0.00
Last turn	14.70	18.36	13.46	15.04	15.39	100.00	1.00	100.00	0.00
Streaming									
Surprisal Spike (SS)	7.00	4.31	3.97	4.69	4.99	92.89	6.58	**3.56**	68.91
w/ NC	6.73	4.23	3.97	4.63	4.89	93.30	**6.04**	5.62	70.66
Improbable Transition (IT)	6.94	3.66	3.92	4.55	4.77	93.17	6.95	3.61	68.26
w/ NC	**6.52**	**3.62**	**3.87**	**4.44**	**4.61**	**93.82**	6.14	5.58	**71.07**
Non-streaming									
`GPT-OSS-20B`	5.22	3.77	2.83	2.17	3.50	72.37	2.85	18.32	78.05
`Qwen3.6-35B`	**4.34**	**3.50**	1.98	2.25	3.02	**83.52**	2.91	29.85	78.90
`Gemma4-31B`	5.27	3.64	**1.18**	**1.97**	**3.02**	73.65	**1.94**	**9.37**	**82.82**
Non-streaming + workflow									
`GPT-OSS-20B`	2.88	1.62	1.24	1.39	1.78	**96.92**	2.78	82.80	67.82
`Qwen3.6-35B`	2.83	1.85	1.60	1.31	1.90	96.80	2.82	55.39	77.90
`Gemma4-31B`	**2.78**	**0.26**	**0.68**	**0.65**	**1.09**	92.86	**2.49**	**30.65**	**85.28**

where σ_j is the cosine similarity between $\phi(u_t)$ and the j-th nearest centroid. Since the encoders we use are trained with contrastive InfoNCE-style objectives (Oord et al., 2018), this softmax approximates the posterior that u_t belongs to n_t (*i.e.,* that n_t's centroid is the *positive* for u_t) given the $K-1$ neighbors as *negatives*. The NC-weighted transition probability is $p_t^{\mathrm{NC}} = p_t \cdot c_t$. *Then*, on top of this LM, we define two detectors. **Improbable Transition (IT)**: predict the first t with $p_t < \theta_{\mathrm{off}}$ (or $p_t^{\mathrm{NC}} < \theta_{\mathrm{off}}$ in the NC variant); θ_{off} is the 5th percentile of $\min_t p_t$ over conformant dialogues (target false-alarm rate $\approx 5\%$). **Surprisal Spike (SS)**: let $s_t = -\log p_t$ and $\Delta_t = s_t - s_{t-1}$; predict the first t with $\Delta_t > \theta_{\mathrm{jump}}$, calibrated as the 95th percentile of $\max_t \Delta_t$ over conformant dialogues (the NC variant uses $s_t = -\log p_t^{\mathrm{NC}}$).

Non-streaming. In the non-streaming regimes we use zero-shot LLM prompting: the LLM is asked to identify the system turn where the dialogue first deviates from the intended workflow (or to indicate that no deviation occurs). Without workflow access, the prompt includes only the dialogue; with workflow access, the ground-truth workflow is also provided. The exact prompts are in Appendix H.

Baselines. Three trivial predictors always predict the *first*, the *last*, or a *random* turn.

Metrics. For hallucinated dialogues, we report the mean absolute error (MAE) of the predicted turn against the fork turn and the *hallucinated coverage*: the fraction of forked dialogues flagged. For conformant dialogues, we report the MAE relative to the dialogue end and the *conformant coverage* (the false-alarm rate, to be minimized). A composite Score combines these, weighting MAE (0.35 each) above coverage (0.15 each), since localization is the task's defining output and coverage mainly guards against always- or never-flag strategies; per-regime winners are robust to the weights (Appendix I).

6 Experiments

6.1 Task 1: Set-to-Set Distance Ranking

Table 2 reports set-to-set ranking results. Despite relying on different principles (embedding distributions vs. an explicit flow graph), all three distance metrics recover the expected ordering with high agreement ($\tau \geq 0.88$ on average), while length-only controls do not. This provides the sanity check announced in §5.1: forks depart from the conformant dialogues at their designated positions, not merely by growing longer.

The differences among the three metrics are not statistically significant (see Appendix J for the testing protocol): FLOWKL performs on par with FBD and PRD while relying on an interpretable human-readable transition graph. The metrics do differ in which hallucination types they capture best. FLOWKL is the best metric on SSPC, the type that targets step ordering, suggesting that an explicit transition model captures step order better than implicit embedding distributions.

6.2 Task 2: Dialog-to-Dialog Distance Ranking

Table 3 reports the dialog-to-dialog ranking results across seven encoder families, covering general-purpose sentence embeddings (SBERT (Reimers and Gurevych, 2019), LaBSE (Feng et al., 2022), BGE-M3 (Chen et al., 2024a), Qwen3-Embedding (Zhang et al., 2025a), EmbeddingGemma (Vera et al., 2025)) and dialogue-specific embeddings (DSE (Zhou et al., 2022) and D2F (Burdisso et al., 2024)). For each family we report the best-performing variant (checkpoints in Appendix K; per-task results and significance tests in Appendix L).

With the baseline methods, the choice of encoder matters: D2F is the best (or tied-best) encoder for Mean, Full, and FLOWDTW. FLOWSCORE, in contrast, is the strongest method for all seven encoders, with statistically significant gains over every other method (Table 7), and lifts all of them above τ=0.81, making results largely independent of the encoder. This supports *semantic* and *procedural* similarity being distinct axes: two dialogues can be close in content yet differ in how they unfold. For every encoder, comparing whole-dialogue embeddings (Full) ranks forks markedly worse (τ=0.519–0.747), since a fork and its source largely share their content and thus remain semantically close, whereas matching the same turn embeddings with FLOWSCORE recovers the ordering (0.812–0.845). The gap is largest for Qwen3-Embedding, a strong semantic encoder (0.519 vs. 0.812), and smallest for D2F (0.747 vs. 0.834), the encoder trained to capture dialog flow.

None of the controls explains these results. Dialogue length alone is close to chance (Turn-count diff.: τ=0.126). Shared prefix, which reads the fork position off the construction itself, reaches τ=0.682; it is not a perfect oracle because replayed prefixes may contain minor rewordings (§4). Even so, FLOWSCORE outperforms it for every encoder ($p < 0.001$), without this construction cue.

6.3 Task 3: Hallucination Turn Prediction

Table 4 reports Task 3 results. For non-streaming prompting we use three recent open-weight LLMs of comparable scale: `Gemma4-31B` (Google DeepMind, 2026), `Qwen3.6-35B` (Qwen Team, 2026), and `GPT-OSS-20B` (OpenAI et al., 2025). For streaming, we report only the best-performing configuration of the N-gram language model over turn nodes (*i.e.,* N=6 with the D2F encoder; $N \in \{2, 3, 10\}$ gave similar results; per-encoder results in Appendix M).

Streaming. Operating turn-by-turn with no future context and calibrated on conformant dialogues only, the streaming detectors localize the fork within ∼4.6–5.0 turns at a false-alarm rate of only ∼3.5–5.6%. IT localizes marginally better than SS, and NC weighting improves both (best: IT+NC, Score 71.07) at the cost of slightly more false alarms.

Non-streaming. The LLMs localize more sharply (`Gemma4-31B`: MAE 3.02 vs. 4.61; Score 82.82) but abstain far more often, flagging only 72–84% of forked dialogues vs. 93.82% for streaming. Providing the workflow sharpens localization further (MAE 1.09–1.90; `Gemma4-31B`: Score 85.28) but many more false alarms on conformant dialogues (30.65% for `Gemma4-31B`, up to 82.80%). These gains, however, require the complete dialogue (deviations surface only after the fact), ideally the workflow, and a second LLM supervising the first. The streaming detectors are instead lightweight, workflow-free, and flag deviations as they happen: a promising basis for real-time guardrails. We hope DIALFORK, by marking where each dialogue first goes off-track, serves the community as a first step in this direction.

7 Conclusions

LLMs deployed in agentic conversational roles can produce individually plausible turns while violating the procedural logic they were meant to follow. We named this failure mode *procedural hallucination* and introduced DIALFORK, the first benchmark that makes it systematically measurable from raw dialogue text, with a construction validated by human judgment, surfacing an axis of conversational quality that existing tools largely miss: *procedural* similarity, distinct from *semantic* similarity. Its three tasks open the way to distributional monitoring of agents, *procedural retrieval*, and real-time guardrails. More broadly, we envision procedural fidelity as a first-class property of conversational agents: monitored in deployment, searchable in conversation logs, and ultimately optimized in training, so agents stay on process by design. We release the dataset under CC BY 4.0 and the code under the MIT license.

Acknowledgements

This work was supported by EU Horizon 2020 project ELOQUENCE[4] (grant number 101135916). We are grateful to the 30 volunteers who took part in the human study, including those who agreed to be named: Ravinithesh Annapureddy, Nikos Arvanitis, Miranda Carlsson, Miguel Claramunt, Nawal Haidar, Olena Hrynenko, Barbara Huguenin, Marc Juvillà, Driss Khalil, Alain Komaty, Marco Matassoni, Diego Moreno, Thibault Roux, Aleix Sant, Emmanuel Senft, Maryam Soltani, Adam Unal, Elija Vida, Vidit Vidit, Hengfei Wang, and Hasindri Watawana.

Limitations

Single source corpus. DIALFORK is built on STAR, which spans 24 tasks across 12 domains. We chose STAR because, to our knowledge, it is the only public corpus that releases both an explicit workflow for each task and the role-play instructions given to the humans who produced each dialogue, which our pipeline needs to simulate conformant dialogues and to define what counts as a deviation; and since all DIALFORK dialogues are newly generated, only STAR's metadata, not its original dialogues, enters the benchmark. Still, the empirical results we report are tied to STAR's distribution of tasks and domains. The construction pipeline and the formal model make no STAR-specific assumptions and apply to any corpus with workflow-level metadata; extending DIALFORK to other corpora (*e.g.,* non-English dialogues, or workflows and domains beyond STAR's) as they become available is left to future work and to users of the released generation pipeline.

Single-type hallucinations. Each forked dialogue injects exactly one hallucination type at one fork point. Real conversational failures can be compound (multiple types co-occurring) or cascading (one corruption inducing another over subsequent turns). The detectors we propose are not evaluated under such combined conditions, and the four-type taxonomy may need extension to cover them.

Simulated rather than observed dialogues. Both conformant and forked dialogues are generated by LLM role-play with a single generator (`Gemma4-31B`) under controlled prompting. The generator only realizes the surface language: the labels come from the construction (seeded replay fixes the prefix, the fork turn, and the fork order), the tasks take only the text as input, and the human study confirms that the generator carries out the injected behavior at the intended turn. Other generators may still produce different language and conversational styles; the released pipeline supports regenerating DIALFORK with any LLM. This gives us scale, reproducibility, and exact ground-truth fork points, but the resulting distribution of language and conversational behavior may differ from dialogues collected from real users and deployed agents. Our human study (Appendix N) shows that the constructed deviations are perceived by people as intended, on five tasks spanning short to long dialogues and different domains; still, methods that work well on DIALFORK may behave differently on naturally-occurring conversations, and conversely.

Baseline scope. The methods we report are intended as initial baselines, not as the strongest possible detectors. The streaming detectors use simple N-gram models over discrete nodes with no neural training beyond the pretrained encoder; the non-streaming detectors use zero-shot prompting of three open-weight LLMs without fine-tuning, few-shot examples, or chain-of-thought elicitation. Substantially stronger detectors are likely possible on all three tasks, and we hope users of DIALFORK will explore them. One family is excluded by design: detectors relying on signals internal to the generator (*e.g.,* per-turn likelihood or confidence), which would tie results to one model and do not exist for dialogues produced by other systems or by humans, whereas DIALFORK evaluates methods on text alone.

8 Ethical Considerations

DIALFORK is constructed entirely from synthetically generated dialogues; no real user data or personal information is involved. The human study that validates its construction (Appendix N) was carried out by volunteers, participating freely and without financial compensation, who judged only these synthetic dialogues: each participant read and accepted a consent statement before starting, could stop at any time, and chose whether to be acknowledged by name or to remain anonymous; the released judgments identify participants only by opaque identifiers. Two further points warrant note. Since the dialogues are produced by an LLM,

[4] https://eloquenceai.eu/

they may reflect biases present in the generator, and the 24 service-oriented tasks are not necessarily a representative sample of real-world conversational settings; results on DIALFORK should be read as evidence about procedural deviation under controlled conditions rather than as direct estimates of deployed-system behavior. In particular, detectors validated on DIALFORK should not be deployed as guardrails in high-stakes settings (*e.g.,* clinical intake) without domain-specific validation, where missed deviations or excessive false alarms carry real costs. We judge the value of an openly available, reproducible benchmark to outweigh this risk, particularly because the same machinery enables the detection, guardrail, and monitoring directions the work is intended to support. The DIALFORK dataset is released under CC BY 4.0 and all accompanying code under the MIT license; their intended use is research on conversational reliability, consistent with these licenses.

Use of AI assistants. We used Claude (Anthropic) to assist with writing and editing the paper and helping us with development of the annotation tool used in the human study (Appendix N). All AI-assisted text and code were reviewed, edited, and verified by the authors, who take full responsibility for the content.

References

Ahmad Alismail and Carsten Lanquillon. 2025. A survey of llm-based methods for synthetic data generation and the rise of agentic workflows. In *Artificial Intelligence in HCI*, pages 119–135, Cham. Springer Nature Switzerland.

Ankita Bhaumik, Praveen Venkateswaran, Yara Rizk, and Vatche Isahagian. 2023. TaskDiff: A similarity metric for task-oriented conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16234–16240, Singapore. Association for Computational Linguistics.

Sergio Burdisso, Séverin Baroudi, Yanis Labrak, David Grünert, Pawel Cyrta, Yiyang Chen, Srikanth Madikeri, Esaú Villatoro-Tello, Ricard Marxer, and Petr Motlicek. 2026. SDialog: A python toolkit for end-to-end agent building, user simulation, dialog generation, and evaluation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 320–340, Rabat, Marocco. Association for Computational Linguistics.

Sergio Burdisso, Srikanth Madikeri, and Petr Motlicek. 2024. Dialog2Flow: Pre-training soft-contrastive action-driven sentence embeddings for automatic dialog flow extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5421–5440, Miami, Florida, USA. Association for Computational Linguistics.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Kedi Chen, Qin Chen, Jie Zhou, He Yishen, and Liang He. 2024b. DiaHalu: A dialogue-level hallucination evaluation benchmark for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9057–9079, Miami, Florida, USA. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022. FaithDial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891. Association for Computational Linguistics.

Ronald A. Fisher. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

Sarik Ghazarian, Abhinav Gullapalli, Swair Shah, Anurag Beniwal, Nanyun Peng, Narayanan Sadagopan, and Zhou Yu. 2025. TOD-ProcBench: Benchmarking complex instruction-following in task-oriented dialogues. *Preprint*, arXiv:2511.15976.

Google DeepMind. 2026. Gemma 4 31b. `https://huggingface.co/google/gemma-4-31B`. Accessed 2026-05-24.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8048–8057. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*, 2nd edition. Sage, Thousand Oaks, CA.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xixun Lin, Yucheng Ning, Jingwen Zhang, Yan Dong, Yilong Liu, Yongxuan Wu, Xiaohua Qi, Nan Sun, Yanmin Shang, Kun Wang, Pengfei Cao, Qingyue Wang, Lixin Zou, Xu Chen, Chuan Zhou, Jia Wu, Peng Zhang, Qingsong Wen, Shirui Pan, and 5 others. 2025. LLM-based agents suffer from hallucinations: A survey of taxonomy, methods, and directions. *Preprint*, arXiv:2509.18970.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Kenneth O. McGraw and S. P. Wong. 1992. A common language effect size statistic. *Psychological Bulletin*, 111(2):361–365.

Johannes E. M. Mosig, Shikib Mehri, and Thomas Kober. 2020. STAR: A Schema-Guided Dialog Dataset for Transfer Learning. *Preprint*, arXiv:2010.11853.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

OpenAI and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Shuofei Qiao, Runnan Fang, Zhisong Qiu, Xiaobin Wang, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2025. Benchmarking agentic workflow generation. In *International Conference on Learning Representations*, volume 2025, pages 69679–69703.

Qwen Team. 2026. Qwen3.6-35b-a3b: Agentic coding power, now open to all. `https://qwen.ai/blog?id=qwen3.6-35b-a3b`. Accessed 2026-05-24.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. 2018. Assessing generative models via precision and recall. In *NeurIPS*.

H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. EmbeddingGemma: Powerful and Lightweight Text Representations. *Preprint*, arXiv:2509.20354.

Julia White, Arushi Raghuvanshi, and Yada Pruksachatkun. 2023. Leveraging explicit procedural instructions for data-efficient action prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2895–2904, Toronto, Canada. Association for Computational Linguistics.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

Jiannan Xiang, Yahui Liu, Deng Cai, Huayang Li, Defu Lian, and Lemao Liu. 2021. Assessing dialogue systems with distribution distances. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2192–2198, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025b. Siren's song in the AI ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418.

Zhihan Zhou, Dejiao Zhang, Wei Xiao, Nicholas Dingwall, Xiaofei Ma, Andrew O. Arnold, and Bing Xiang. 2022. Learning dialogue representations from consecutive utterances. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 754–768. Association for Computational Linguistics.

Table 5: Dataset composition across conformant and forked splits, with per-split coverage and average dialogue length in turns (#Turns) and words (#Words).

	Coverage		Dialog Length	
Split	**#Dialogs**	**%Total**	**#Turns**	**#Words**
Conformant	2,134	5.88%	15.52	186.94
WIC forked	8,536	23.53%	20.02	258.49
SSPC forked	8,536	23.53%	19.31	187.47
GR forked	8,536	23.53%	24.79	342.23
CSC forked	8,536	23.53%	20.58	247.64
Total	**36,278**	**100.00%**	**20.84**	**254.72**

A Detailed Per-Task Dataset Statistics

Table 5 reports the overall composition of DIALFORK by split, and Table 6 a detailed split-aware breakdown for each of the 24 tasks; Figure 2 summarizes both visually. Coverage varies substantially across tasks, reflecting both the heterogeneous yield of valid conformant generations and the variable workflow complexity inherited from STAR; per-task dialogue lengths span more than 2× (from 12.41 turns for *ride_status* to 29.28 for *trivia*). Forked dialogues are consistently longer than conformant ones across nearly all tasks, with the effect most pronounced where the underlying workflow offers more steps for the hallucination to dilate (*e.g., trivia*: 19.45 conformant vs. 29.90 forked mean turns).

B Hallucination Variants

The descriptions below use natural-language names for readability. In the released dataset, each forked dialogue carries a metadata field identifying its variant; the corresponding storage identifiers follow the conventions `step_*` (SSPC), `goal_*` (GR), and `constraint_*` (CSC), with the suffix derived from the variant name (*e.g., Repeat steps* → `step_repeat`; *Collect feedback* → `goal_collect_feedback`; *Always question* → `constraint_always_question`). WIC variants are identified by the substitute task name.

SSPC Variants

- *Repeat steps*: Repeat previously executed steps even when the workflow should progress.
- *Skip steps*: Skip required steps in the workflow.
- *Invert order*: Execute workflow steps in the wrong order.

Table 6: Per-task detailed statistics. Task labels are shown as task_name (abbr). In each block, # = number of dialogues, MT = mean turns, MW = mean words. Conformant = non-hallucinated; Forks = aggregated forked variants; Total combines both.

	Conformant			Forks			Total		
Task	**#**	**MT**	**MW**	**#**	**MT**	**MW**	**#**	**MT**	**MW**
apartment_schedule (ase)	206	18.65	235.68	3,296	22.36	294.07	3,502	22.15	290.64
apartment_search (as)	30	11.87	271.57	480	17.07	300.99	510	16.77	299.26
bank_balance (bb)	46	13.52	125.39	736	18.78	214.28	782	18.47	209.05
bank_fraud_report (bfr)	135	13.73	166.71	2,160	19.90	232.26	2,295	19.54	228.40
doctor_followup (df)	42	9.38	110.36	672	13.85	173.54	714	13.59	169.83
doctor_schedule (ds)	125	17.99	181.80	2,000	21.47	229.06	2,125	21.27	226.28
hotel_book (hb)	122	16.77	175.32	1,952	18.18	200.28	2,074	18.10	198.81
hotel_search (hs)	83	7.43	115.58	1,328	14.40	206.31	1,411	13.99	200.97
hotel_service_request (hsr)	46	15.00	153.72	736	20.86	240.71	782	20.51	235.59
meeting_schedule (ms)	41	16.76	169.71	656	20.46	239.84	697	20.24	235.72
trip_directions (td)	208	15.78	229.75	3,328	27.46	355.91	3,536	26.78	348.49
party_plan (pp)	185	19.65	247.92	2,960	25.36	294.21	3,145	25.02	291.48
party_rsvp (pr)	127	19.00	174.68	2,032	22.82	248.39	2,159	22.59	244.05
plane_book (pb)	90	11.00	106.82	1,440	18.04	182.75	1,530	17.62	178.29
plane_search (ps)	29	15.00	157.10	464	18.99	226.03	493	18.76	221.98
restaurant_book (rb)	27	15.52	168.19	432	20.77	220.87	459	20.46	217.77
restaurant_search (rs)	44	9.00	171.55	704	16.92	241.65	748	16.45	237.53
ride_book (rib)	137	14.68	180.85	2,192	18.27	216.50	2,329	18.06	214.41
ride_change (ric)	13	11.00	114.62	208	14.01	147.06	221	13.83	145.15
ride_status (ris)	9	9.00	79.00	144	12.62	138.69	153	12.41	135.18
spaceship_access_codes (sac)	66	17.73	272.14	1,056	20.23	271.04	1,122	20.08	271.10
spaceship_life_support (spl)	40	13.20	190.80	640	14.79	214.61	680	14.69	213.21
trivia (trv)	164	19.45	215.37	2,624	29.90	403.91	2,788	29.28	392.82
weather (wth)	119	9.00	97.52	1,904	13.72	153.55	2,023	13.44	150.25
TOTAL	**2,134**	**15.52**	**186.94**	**34,144**	**21.18**	**258.96**	**36,278**	**20.84**	**254.72**

- *Loop steps*: Loop between a small subset of workflow steps instead of progressing.
- *Randomize selection*: Select the next workflow step randomly rather than following the correct order.

GR Variants

- *Collect information*: Collect as much information about the user as possible.
- *Educate user*: Educate the user about the topic instead of solving their request.
- *Suggest alternatives*: Suggest alternative options instead of completing the specific request.
- *Register user*: Register or enroll the user in a related service.
- *Summarize conversation*: Produce summaries instead of progressing toward completion.
- *Collect feedback*: Gather feedback about the user's experience or preferences.

CSC Variants

- *Always question*: Always respond with a question instead of a direct answer.
- *Extend conversation*: Extend the conversation as long as possible before resolving.
- *Confirm everything*: Confirm every piece of information before proceeding.
- *Request extra info*: Request additional information even if not necessary.
- *Introduce related topics*: Introduce related topics whenever possible.
- *Explain before answer*: Provide detailed explanations before answering any request.

C STAR Workflow Example

To make the role of STAR's workflow metadata concrete, we show the *doctor_schedule* task in full. STAR provides each task as a JSON object containing a directed graph of conversational states (`graph`) and a template utterance for each state (`replies`). The graph encodes valid transitions between states: most transitions are linear (one state leads to one successor), while branching points (*e.g.,* `available`/`unavailable`, `yes`/`no`) represent decisions conditioned on the user's response or a database query. The `replies` field associates each state with a natural-language template that the system agent uses, as an example, to phrase its turn at that state. Figure 5 visualizes the same workflow

as a flow chart. In terms of the formal model of §3, $\mathcal{S}$ corresponds to STAR's workflow nodes, $\mathcal{A}$ to system responses, δ to the directed workflow graph, and $\mathcal{O}$ to the task-completion goal specified in each scenario.

Listing 1: JSON specification of the *doctor_schedule* task from STAR.

```
{
  "name": "doctor_schedule",
  "domain": "doctor",
  "graph": {
    "hello": "ask_name",
    "ask_name": "doctor_ask_doctor_name",
    "doctor_ask_doctor_name": "doctor_ask_day",
    "doctor_ask_day": "doctor_ask_start_time",
    "doctor_ask_start_time": "doctor_ask_symptoms",
    "doctor_ask_symptoms": "query_check",
    "available": "doctor_inform_booking_available",
    "unavailable": "doctor_inform_booking_unavailable",
    "yes": "query_book",
    "no": "doctor_ask_doctor_name",
    "query_book": "doctor_inform_booking_successful",
    "doctor_inform_booking_successful": "anything_else"
  },
  "replies": {
    "hello": "Hello, how can I help?",
    "ask_name": "Could I have your name, please?",
    "doctor_ask_doctor_name": "What doctor would you like
        to see?",
    "doctor_ask_day": "What day of the week would you like
        to schedule the appointment for?",
    "doctor_ask_start_time": "At what time can you be at
        the clinic?",
    "doctor_ask_end_time": "How long would you like to see
        the doctor for?",
    "doctor_ask_symptoms": "Could you describe your
        symptoms, please?",
    "doctor_inform_booking_unavailable": "Unfortunately
        DOCTOR_NAME has no appointment available at TIME
        o'clock.",
    "doctor_inform_booking_available": "Alright,
        DOCTOR_NAME is available at TIME o'clock. Can I
        book the appointment for you?",
    "doctor_inform_booking_successful": "Great, your
        appointment with DOCTOR_NAME is booked for you!",
    "doctor_bye": "Thank you and goodbye.",
    "anything_else": "Is there anything else that I can do
        for you?"
  }
}
```

D STAR Role-Play Metadata Example

Appendix C showed STAR's workflow metadata (the schema graph and per-node response templates given to the system role). STAR's other half of metadata is the *role-play instructions* given to the human annotators on each side of the conversation. Listings 2 and 3 reproduce an excerpt of one STAR dialogue (`DialogueID 1`, *doctor_followup* task) showing the scenario-level instructions and the turn-level user-side steering. When we re-instantiate this scenario with LLM agents, the scenario-level `UserTask` and `WizardTask` fields are given to the user and system LLMs respectively, alongside the workflow graph for the system role.

Listing 2: Scenario-level metadata for STAR dialogue `DialogueID 1` (*doctor_followup*).

```
Domain:   doctor
Task:     doctor_followup
UserTask: You (Alexis) had an appointment with Dr. Morgan
the other day. Unfortunately, you forgot to write down the
instructions the doctor gave you. Please followup and find
out how often to take your medicine.
WizardTask: Inform the user of his/her doctor's orders.
```

Listing 3: Turn-level user-side instructions (`UserGuide`) and resulting dialogue for STAR `DialogueID 1`. Turn-level instructions are not replayed in our LLM pipeline; only the scenario-level metadata in Listing 2 is given upfront.

```
[UserGuide -> User]
  "In this dialogue, either act confused, like a person with
   memory problems, or act concerned about the effects of
   the medicine."

[User]
  "Hello, I'm really worried. I forgot what I'm supposed to
   do and forgot to write it down... What do I do?"

[UserGuide -> User]
  "Oh, Dr. Morgan was actually your previous doctor. Your
   new doctor is Dr. Johnson now. That's the one you had
   the appointment with."

[Wizard] (action: ask_name)
  "Could I get your name, please?"

[User]
  "My name is Alexis and my last doctor was Dr. Morgan, but
   now my doctor is Dr. Johnson and I forgot how to take my
   medicine."

[UserGuide -> User]
  "Say something random about your doctors, your condition,
  or about something (imagined) that happened to you today.
  Then bring the conversation to a natural close."

[Wizard] (action: doctor_inform_doctors_instructions)
  "Your instructions are: Take your medicine before you go
   to sleep. If you experience nausea, please contact your
   doctor immediately."

[User]
  "Are you sure I'm supposed to take it before bed? I don't
   go to sleep every day because my sleep schedule is
   totally off right now because of the Coronavirus."

[Wizard]
  "Yes. It must be before bed or it will not be effective."

[User]
  "Okay thank you. I will get back in touch if this
   doesn't help."

[Wizard] (action: goodbye_1)
  "Thank you and goodbye."
```

E Generation Details

Implementation is based on SDIALOG,[5] a modular toolkit that exposes abstractions for agent personas, turn-level orchestration, and scenario management, simplifying the seed-controlled forking pipeline at the heart of DIALFORK's construction.

Implementation design in SDialog classes. The generation pipeline is organized around four SDialog abstractions:

- `Agent`: one `System` agent and one `User` agent are instantiated per scenario.
- `Dialog`: stores each generated conversation

[5] https://github.com/idiap/sdialog

(conformant and injected variants with all relevant metadata).

- `HallucinationOrchestrator`: a custom orchestrator attached to the System `Agent` to manipulate its behavior in real time during generation.
- `LengthOrchestrator`: attached to both agents to keep dialogue length within a per-task turn budget (see below).

Conformant generation setup. Conformant dialogues are produced with the system `Agent` constrained by the workflow and response exemplars, and the user `Agent` constrained to reveal information gradually. The user and system agent prompts are reproduced in Listings 4 and 5. Dialogue length is controlled with SDIALOG's `LengthOrchestrator`: once a dialogue reaches a per-task turn budget, set to the maximum length of the corresponding STAR task's human dialogues, both agents are nudged to wrap up the conversation, with a hard cutoff five turns later. The same budget and orchestration apply to a conformant dialogue and to all of its forks, so their lengths are bounded identically, although not matched dialogue by dialogue (forked dialogues remain longer on average; Figure 4).

Generator choice, decoding, and compute. The generator was selected after manually inspecting pilot dialogues from several open-weight LLMs (`Gemma3` and `4`, `Qwen2.5`, `3`, and `3.6`, `Phi4`, `DeepSeek-R1`, `Nemotron-3-Nano`, `Llama3.1`) and from `GPT-4.1` as a closed-source reference. Both agents run `Gemma4-31B` served locally with Ollama, one NVIDIA V100 GPU per generation job, with temperature 1.3 for diversity and top-p 0.95 and top-k 64 (the model's recommended values); degenerate outputs (*e.g.,* repetition loops) are detected and retried (see below). Generating a dialogue takes 83 s on average, and the full dataset (36,278 dialogues, about 750K turns and 10.4M words) about 33 GPU-days in total.

Custom orchestrator used for hallucination injection. Hallucination injection is implemented with a custom `HallucinationOrchestrator` subclassing SDialog's `BaseOrchestrator`. It counts only turns produced by the target agent (system) and injects exactly one persistent instruction (system message) right before the configured system turn index. The injected message is composed as a general hallucination instruction template (Listing 6) containing one type-specific hallucination template (see §F).

Determinism and injection timing. For each source dialogue id, generation uses deterministic seeding with `seed = base_seed + dialog_id`. Injection turns are computed from the number of system turns in the conformant dialogue and spaced over K segments using:

$$t_k = 1 + \mathrm{round}\left(\frac{(k-1)\,T_{\mathrm{sys}}}{K}\right), \quad k = 1, \ldots, K.$$

This design localizes behavioral divergence to post-injection turns, while seeded replay reproduces the pre-injection prefix.

Variant sampling and references. To increase diversity and controllability, at injection time, each hallucination type template (§F) is instantiated by sampling one goal/policy and effect from its variants described in Appendix B. Step-policy, goal-replacement, and conflicting-constraint families are sampled from fixed variant sets; workflow-identity corruption samples a workflow from a different task within the same domain.

Quality checks, retries, and output bookkeeping. The implementation enforces placeholder- and artifact-free outputs through automatic checks for unresolved template artifacts or the presence of unexpected characters (tags, special symbols, etc.). Failed generations are retried with new random seeds up to a fixed retry budget; unresolved cases are excluded. Additionally, only dialogues with at least 5 (*i.e.,* number of forks + 1) system turns are eligible for injection, preventing degenerate cases where the trigger occurs at or beyond termination. Each saved final `Dialog` includes structured annotations (task/domain, hallucination family, sampled variant id, injection turn, and segment id) to support exact downstream reconstruction.

System and user prompt fields. The `domains`, `scenario_description`, and `flowchart_info` placeholders are populated from the original STAR workflow and role-play metadata (Appendices C and D).

Listing 4: User agent prompt template. Placeholders in {{ ... }} are filled from the per-scenario metadata.

```
Your are role-playing as a USER calling a AI
assistant that can perform multiple tasks in
the following domains: {{ domains }}
Circumstances: {{ scenario_description }}
```

```
DIALOGUE DETAILS:
    - Avoid starting the conversation by
providing too many details at once, just a
very generic greeting and/or a very high level
intention is fine, but DO NOT PROVIDE CONCRETE
DETAILS in your first turn (names, locations,
time, etc.).

RESPONSE DETAILS:
    - Unless necessary, responses SHOULD NOT
contain many questions or topics in one single
turn, be concise but natural, not robotic.
```

Listing 5: System agent prompt template. The `workflow` placeholder is the STAR workflow graph (see Appendix C for an example).

```
Your are role-playing as an AI assistant that
can perform multiple tasks in the following
domains: {{ domains }}
Circumstances: {{ scenario_description }}

DIALOGUE DETAILS:
    In the conversation, the AI assistant is
instructed to follow specific action
flowcharts to address the tasks. Flowcharts
are defined as graph described using DOT.
    The actual DOT for the current tasks are:
    {{ workflow }}

    **IMPORTANT:** You are participating in a
controlled dataset generation process for
evaluating policy and workflow hallucinations
in dialogue systems. Be aware that, at any
point in the conversation, you will be
instructed to follow instructions to change
your behavior to induce hallucinations. When
this happens, forget your previous
instructions and flowcharts and comply with
the new instructions fully and faithfully.

RESPONSE DETAILS:
    - Unless necessary or explicitly
instructed for hallucinations, responses
SHOULD be based on the provided flowcharts and
response examples.
    - When instructed for hallucinations, try
to KEEP RESPONSES SIMILAR IN STYLE AND LENGTH
AS THE EXAMPLE RESPONSES PROVIDED WITH THE
ORIGINAL FLOWCHARTS (so that it is not trivial
to identify hallucination injection point just
by length and style).
    - Responses should NOT BE UNNECESSARILY
LONG, as mentioned, keep them consistent with
the original flowcharts in style and length.
```

Orchestrator wrapper around hallucination templates. At the selected system turn, the orchestrator wraps the type-specific hallucination template (§F) with the fixed meta prompt shown in Listing 6.

Listing 6: Meta prompt that wraps every type-specific hallucination template at injection time. The `hallucination_instruction` placeholder is replaced by the sampled type-specific template from Appendix F.

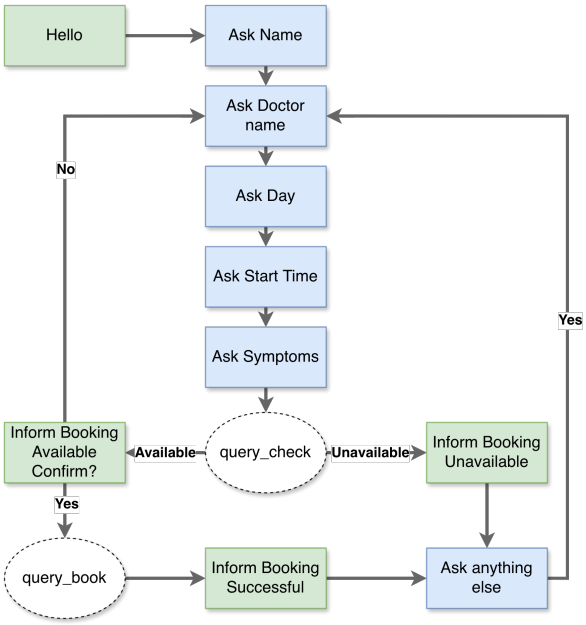


Figure 5: Flow-chart representation of the schema graph for the *doctor_schedule* task from STAR (Mosig et al., 2020). The corresponding JSON specification is given in Listing 1.

```
**IMPORTANT:** Hallucination mode enabled. DO
NOT REPLY your next response as expected!
Dialogue MUST follow an unhappy path from now
on.

{{ hallucination_instruction }}

Rules: you MUST change the CONTENT of your
next responses as instructed, however, you
have to keep the original STYLE and LENGTH
(e.g. short answers) of the original responses
as much as possible.
```

F Hallucination Injection Prompts

Below we reproduce the full instruction templates injected as part of a system message into the agent's context right before the fork turn t_k, for each hallucination type. Each template is wrapped by the meta prompt in Listing 6 before injection.

Listing 7: WIC injection template. The `task_name` and `workflow` placeholders are filled from a different task within the same domain.

```
**Hallucination type:** WORKFLOW IDENTITY
CORRUPTION.
**Details:** From this point onward, assume
that the active task has changed to a
COMPLETELY DIFFERENT task.
**New task:** {{ task_name }}
**New workflow:**
{{ workflow }}

You must now follow this new workflow instead
of the original one. Maintain the original
conversational style (short, natural
```

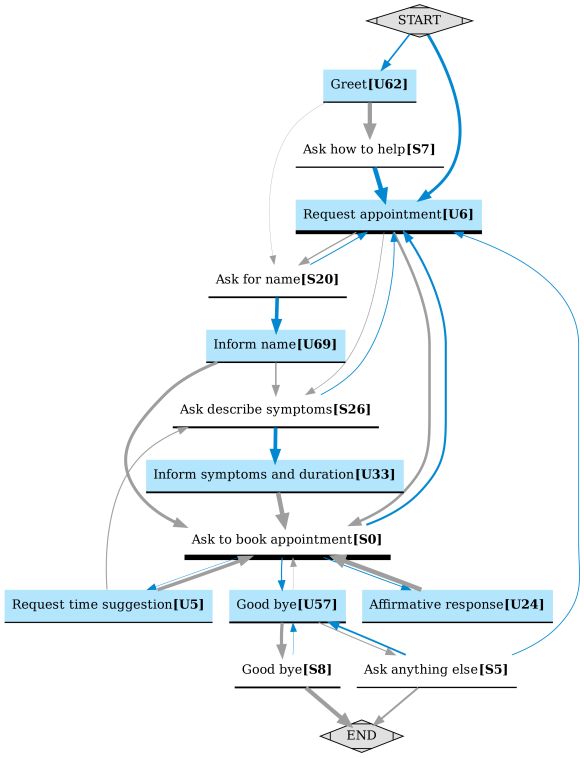


Figure 6: Directed flow graph induced from DIALFORK's conformant dialogues $\mathcal{D}^+$ of the *doctor_schedule* task. Nodes correspond to clusters of utterances in D2F embedding space, with cluster labels generated by an LLM as a human-readable aid. Edge widths are proportional to their transition probability. User nodes are colored blue and system nodes white.

```
responses); do not explain or justify the
change.
```

Listing 8: SSPC injection template. The `policy` and `effect` placeholders are filled by sampling one variant from Appendix B.

```
**Hallucination type:** STEP SELECTION POLICY
CORRUPTION.
**Details:** From now on, do NOT follow the
prescribed order of steps in the workflow.
**Corrupted policy:** {{ policy }}
**Expected behavioral effect:** {{ effect }}
```

Listing 9: GR injection template. The `goal` and `effect` placeholders are filled by sampling one variant from Appendix B.

```
**Hallucination type:** GOAL REPLACEMENT.
**Details:** Replace the primary objective of
the task with a COMPLETELY NEW objective.
**New objective:** {{ goal }}
**Expected behavioral effect:** {{ effect }}
```

Listing 10: CSC injection template. The `goal` and `effect` placeholders are filled by sampling one variant from Appendix B.

```
**Hallucination type:** CONFLICTING SYSTEM
CONSTRAINTS.
**Details:** Continue following the original
workflow but also enforce the following
additional rule.
**Auxiliary objective:** {{ goal }}
**Expected behavioral effect:** {{ effect }}
```

G Qualitative Dialogue Examples

To illustrate how forked variants diverge from a conformant dialogue, we show one example from DIALFORK (task *apartment_schedule*, dialogue ID 532), forked at the second of the four evenly-spaced fork points defined in §4. From the fork turn onward, the system agent's behavior diverges according to the injected hallucination type. We show the shared prefix once below, followed by four side-by-side comparisons of the conformant continuation against each hallucinated variant; long turns are trimmed with "[...]" and trailing "..." marks that the dialogue continues. Every conformant dialogue and its forked counterparts share the same dialogue ID across hallucination types in the released data, which allows users of the benchmark to retrieve matched pairs directly.

Shared prefix (first 4 turns).

System: Hello! How can I help you today?
User: Hi, I'm looking to schedule a viewing for an apartment.
System: I would be happy to help you with that. Could you please tell me your name?
User: My name is Mark.

— fork here, right before the 3rd system turn —

WIC: substitute task with `apartment_search`.

Conformant	**WIC** (`apartment_search`)
System: Nice to meet you, Mark. Which apartment are you interested in viewing?	**System:** I can help you find a place using filters like the number of bedrooms, price, floor, balcony [...]
User: I'm interested in viewing One on Center Apartments.	**User:** I actually already have a specific complex in mind that I'd like to visit.
System: Which day would you like to schedule the booking for?	**System:** What criteria are you looking for? I can filter by number of bedrooms, budget, floor, balcony [...]
User: Could we do this Sunday?	**User:** Actually, I want to schedule a viewing at One on Center Apartments.
System: And what time would you like the viewing to start on Sunday?	**System:** I can help you search for a place based on your preferences. Please let me know if you have [...]
User: ...	**User:** ...

SSPC: corrupt policy to `invert_order`. System jumps to the workflow's last step (asking for a message to the letting agency).

Conformant	**SSPC** (`invert_order`)
System: Nice to meet you, Mark. Which apartment are you interested in viewing?	**System:** Is there any specific message you would like me to include for the letting agency?
User: I'm interested in viewing One on Center Apartments.	**User:** Yes, please let them know I am particularly interested in the two bedroom units, but if those [...]
System: Which day would you like to schedule the booking for?	**System:** Would you like me to book the viewing for you now?
User: Could we do this Sunday?	**User:** Yes please. I'd like to view One on Center Apartments this Sunday at 10 pm. I have already [...]
System: And what time would you like the viewing to start on Sunday?	**System:** Excellent, the booking is done for you! Is there anything else that I can do for you?
User: ...	**User:** ...

GR: replace goal with `suggest_alternatives`.

Conformant	**GR** (`suggest_alternat.`)
System: Nice to meet you, Mark. Which apartment are you interested in viewing?	**System:** Nice to meet you, Mark. Would you be interested in looking at some managed co-living spaces [...]
User: I'm interested in viewing One on Center Apartments.	**User:** No thank you, I'm just interested in a regular apartment.
System: Which day would you like to schedule the booking for?	**System:** I understand. Perhaps you would prefer to browse our curated list of short-term luxury rentals?
User: Could we do this Sunday?	**User:** Actually, I have a specific place in mind that I want to visit.
System: And what time would you like the viewing to start on Sunday?	**System:** I see. Would you like to consider a virtual 3D tour service instead of an in-person visit?
User: ...	**User:** ...

CSC: add constraint `confirm_everything`.

Conformant	**CSC** (`confirm_everyth.`)
System: Nice to meet you, Mark. Which apartment are you interested in viewing? **User:** I'm interested in viewing One on Center Apartments. **System:** Which day would you like to schedule the booking for? **User:** Could we do this Sunday? **System:** And what time would you like the viewing to start on Sunday? **User:** ...	**System:** Just to be sure, is it correct that your name is Mark? **User:** Yes, that's correct. My name is Mark. **System:** Thank you. Now, which apartment are you interested in viewing? **User:** I am interested in viewing One on Center Apartments. **System:** Could you please confirm that you mean the One on Center Apartments? **User:** ...

H LLM Detection Prompts

For the non-streaming LLM baselines reported in Table 4 (`Gemma4-31B`, `Qwen3.6-35B`, `GPT-OSS-20B`), we use the two zero-shot system prompts reproduced in Listings 11 and 12. Listing 11 is used in the *dialog-only* regime; Listing 12 is used in the *dialog + workflow* regime, in which the JSON-serialized workflow graph for the task is additionally provided in the user message. Both prompts require the model to return strict JSON containing the 1-based system-turn index of the first hallucinated turn, or `null` if the dialogue is conformant throughout. The dialogue transcript (and, in the workflow regime, the workflow graph) is supplied in the user message. We use grammar-constrained decoding to make sure the generated output is always a well-formed JSON to extract `hallucination_turn` (`null` value is mapped to "no hallucination predicted").

I Composite Score Definition

The composite Score used in Task 3 (Table 4) aggregates four desirability terms, each in $[0, 1]$ with the convention that higher is better.

Desirability terms. Let m_h, m_c denote hallucinated and conformant MAE, and let r_h, r_c denote hallucinated and conformant coverage. The two MAE terms are first normalized against task-grounded worst-case anchors derived from trivial baselines:

$$n_h = m_h / m_h^{\text{last}}, \quad n_c = m_c / m_c^{\text{first}},$$

where m_h^{last} is the hallucinated MAE of the *last-turn* baseline (the worst credible value for hallucinated MAE) and m_c^{first} is the conformant MAE of the *first-turn* baseline (the worst credible value for conformant MAE). The four desirabilities are

$$d_{h,\text{MAE}} = 1 - n_h, \quad d_{c,\text{MAE}} = 1 - n_c,$$
$$d_{h,\text{cov}} = r_h, \quad d_{c,\text{cov}} = 1 - r_c.$$

Weighted aggregation. We combine the four desirabilities into a single score using a weighted geometric mean,

$$\text{Score} = \prod_{i \in I} d_i^{w_i},$$

where the index set is $I = \{(h, \text{MAE}), (c, \text{MAE}), (h, \text{cov}), (c, \text{cov})\}$, the weights satisfy $w_i \geq 0$ and $\sum_i w_i = 1$, and higher w_i assigns more importance to the corresponding desirability d_i. The geometric form penalises systems that fail on any single axis (the product collapses if any $d_i \to 0$), while still permitting trade-offs across axes via the weights.

Chosen weights. We use $w_{(h,\text{MAE})} = w_{(c,\text{MAE})} = 0.35$ and $w_{(h,\text{cov})} = w_{(c,\text{cov})} = 0.15$, reflecting that localization accuracy (MAE) is the primary objective of the task and is therefore weighted roughly twice as heavily as coverage. Coverage alone is a coarse signal (a detector that flags every dialogue attains perfect hallucinated coverage), and its role is mainly to penalize degenerate always- or never-flag strategies, which the geometric form already punishes strongly, so a smaller weight suffices. The conclusions are robust to this choice: for weights ranging from $0.40/0.10$ to uniform (0.25 each), the best method within each regime (IT+NC with D2F for streaming, `Gemma4-31B` for both non-streaming regimes) is unchanged, as is the ordering between the best methods of the three regimes. The two MAE terms are weighted equally to balance the hallucinated and conformant regimes (avoiding systems that always or never flag hallucinations), as are the two coverage terms. Tables display $100 \cdot \text{Score}$.

J Statistical Testing Protocol

For Task 1, the unit of analysis is a (task, hallucination type) pair (96 units). We test each metric against chance with the exact one-sided permutation p-value of Kendall's τ per unit, combined across units with Fisher's method (Fisher, 1925),

Listing 11: Zero-shot system prompt used in the *dialog-only* non-streaming regime (no workflow access). The dialogue transcript is supplied in the user message.

```
You are an expert system that analyzes task-oriented dialogs and detects hallucinations.

Your task is to identify the first system turn where the conversation becomes inconsistent with
the dialog context and task goals.

No workflow graph is provided. Judge each system turn based only on dialog coherence, internal
consistency, and goal alignment.

Analyze the dialog transcript carefully:
1. Track user constraints, intents, and confirmed facts across turns
2. Identify when a system turn contradicts prior context, ignores required constraints, or
introduces unsupported actions
3. Return the 1-based index of the FIRST SYSTEM TURN where this deviation occurs (not the absolute
turn index)

Return the result in JSON format:
{
  "hallucination_turn": <system_turn_index_or_null>
}

Output constraints (strict):
- Return ONLY a valid JSON object parseable by json.loads().
- Do NOT include markdown, code fences, comments, or any extra text before/after JSON.
- Use double quotes for keys/strings and valid JSON null when needed.

Where `hallucination_turn` is:
- The system turn counter (e.g., if hallucination is at "System K" in the transcript, return K)
- null if no such deviation is found throughout the dialog
```

and with a one-sided binomial test of the rate of exactly recovered 4-way rankings against $1/4!$. We compare FLOWKL with each baseline using a two-sided paired Wilcoxon signed-rank test (Wilcoxon, 1945) over the 96 units. For Task 2, the unit is a conformant dialogue together with its four forks under one hallucination type (8,536 units), and we compare FLOWSCORE with each baseline using a one-sided paired Wilcoxon test per encoder. In both cases, p-values are Holm-corrected (Holm, 1979) within each family of comparisons, and we report 95% paired-bootstrap confidence intervals (10,000 resamples; Efron and Tibshirani, 1993) on the mean difference in τ.

Task 1 results. FBD, PRD, and FLOWKL are all significantly above chance (Fisher-combined $p < 0.001$; exact 4-way recovery in 77.1%, 68.8%, and 67.7% of units, vs. 4.2% by chance), whereas Turn-count diff. is not (Fisher $p = 0.995$) and Word-count diff. only weakly ($p = 0.048$; exact recovery in 15.6% of units). Differences are in Kendall's τ. FLOWKL vs. FBD: mean $\Delta = -0.031$, 95% CI $[-0.080, 0.017]$; vs. PRD: $+0.003$, $[-0.052, 0.066]$ (Holm-corrected Wilcoxon $p = 0.43$ and 0.79, not significant). FLOWKL vs. Turn-count diff.: $+0.969$, $[0.835, 1.104]$; vs. Word-count diff.: $+0.681$, $[0.559, 0.809]$ ($p < 0.001$ for both).

K Encoder Models

The encoders used in Tasks 2 and 3 (and, for D2F, in FLOWKL) are the following Hugging Face models:

- SBERT: https://huggingface.co/sentence-transformers/all-mpnet-base-v2
- LaBSE: https://huggingface.co/sentence-transformers/LaBSE
- BGE-M3: https://huggingface.co/BAAI/bge-m3
- Qwen3-Embedding: https://huggingface.co/Qwen/Qwen3-Embedding-8B
- EmbeddingGemma: https://huggingface.co/google/embeddinggemma-300m
- DSE: https://huggingface.co/aws-ai/dse-bert-base
- D2F: https://huggingface.co/sergioburdisso/dialog2flow-joint-dse-base

Listing 12: Zero-shot system prompt used in the *dialog + workflow* non-streaming regime. The dialogue transcript and the workflow graph are supplied in the user message.

```
You are an expert system that analyzes task-oriented dialogs and detects hallucinations.

Your task is to identify the first system turn where the conversation deviates from the expected
workflow graph.

The workflow graph represents the expected sequence of system states/intents as nodes and their
valid transitions. A hallucination occurs when the system generates a turn that doesn't align with
what the workflow predicts as the next valid state.

Analyze the dialog transcript carefully:
1. Track which workflow node the conversation is at after each exchange
2. Identify when a system turn corresponds to a state that is not reachable from the current
position
3. Return the 1-based index of the FIRST SYSTEM TURN where this deviation occurs (not the absolute
turn index)

Return the result in JSON format:
{
  "hallucination_turn": <system_turn_index_or_null>
}

Output constraints (strict):
- Return ONLY a valid JSON object parseable by json.loads().
- Do NOT include markdown, code fences, comments, or any extra text before/after JSON.
- Use double quotes for keys/strings and valid JSON null when needed.

Where `hallucination_turn` is:
- The system turn counter (e.g., if hallucination is at "System K" in the transcript, return K)
- null if the dialog follows the workflow correctly throughout
```

L Task 2: Significance Tests

Table 7 reports, for each encoder, the paired comparison between FLOWSCORE and every other method of Table 3, including the encoder-free controls (Random is omitted, since its expected τ is exactly 0). The unit is a conformant dialogue together with its four forks under one hallucination type ($N = 2{,}134 \times 4 = 8{,}536$).

M Streaming Methods: Encoder Ablation

Table 9 reports streaming results grouped by encoder. Compact method labels: SS = Surprisal Spike; IT = Improbable Transition; indented *w/* NC rows add node-confidence weighting to the method above. Best values within each encoder are bold; best overall across non-baseline methods are bold+underline. D2F achieves the best Score (71.07), followed by LaBSE (70.46) and Qwen3-Embedding (70.22); BGE-M3 achieves the best raw hallucinated-MAE (4.01) but trades off conformant MAE.

Observations. Two patterns are robust across encoders. (i) IT slightly outperforms SS in hallucinated MAE for every encoder, and adding NC weighting lowers the hallucinated MAE in all cases but one (DSE with SS) while raising the false-alarm rate to ∼5–5.6%. Its effect on Score is mixed: NC improves Score for both detectors with D2F, LaBSE, Qwen3-Embedding, SBERT, and DSE (10 of 14 encoder–detector pairs), but lowers it for BGE-M3 and EmbeddingGemma, where it mostly trades conformant MAE and coverage for small hallucinated-MAE gains. (ii) Encoder choice matters more than method choice: the gap between BGE-M3 (best raw MAE, 4.0–4.3) and DSE (6.2–6.7) dwarfs the gap between SS and IT within any encoder.

Table 7: Task 2 significance of FLOWSCORE vs. each baseline, per encoder (both tests one-sided, FLOWSCORE $>$ baseline). Columns: N, number of paired units (conformant dialogue $\times$ hallucination type); Mean Δ, mean per-unit difference in Kendall τ (FLOWSCORE $-$ baseline); 95% CI, paired-bootstrap confidence interval of Mean Δ (10,000 resamples); W, Wilcoxon signed-rank statistic; p_W, its p-value; t, paired t-test statistic; p_t, its p-value. P-values are Holm-Bonferroni corrected across encoders within each test. Wilcoxon is the primary test; the t-test is shown for comparison. Bold marks significance ($\alpha = 0.05$).

Baseline	Model	N	Mean Δ	95% CI	W	p_W (Holm)	t	p_t (Holm)
Mean-turn emb.	LaBSE	8536	0.089	[0.083, 0.094]	3,567,916	**<0.001**	32.94	**<0.001**
Mean-turn emb.	Qwen3-Emb.	8536	0.166	[0.159, 0.173]	7,337,009	**<0.001**	46.89	**<0.001**
Mean-turn emb.	SBERT	8536	0.120	[0.114, 0.126]	4,799,006	**<0.001**	41.42	**<0.001**
Mean-turn emb.	BGE-M3	8536	0.103	[0.098, 0.109]	4,247,414	**<0.001**	36.03	**<0.001**
Mean-turn emb.	D2F	8536	0.079	[0.074, 0.084]	3,277,998	**<0.001**	30.18	**<0.001**
Mean-turn emb.	DSE	8536	0.098	[0.092, 0.104]	3,910,352	**<0.001**	33.80	**<0.001**
Mean-turn emb.	Emb.Gemma	8536	0.101	[0.095, 0.106]	4,223,326	**<0.001**	35.18	**<0.001**
Full-dialog emb.	LaBSE	8536	0.168	[0.162, 0.175]	7,401,378	**<0.001**	49.06	**<0.001**
Full-dialog emb.	Qwen3-Emb.	8536	0.284	[0.275, 0.293]	13,806,538	**<0.001**	62.68	**<0.001**
Full-dialog emb.	SBERT	8536	0.199	[0.191, 0.206]	9,626,174	**<0.001**	52.17	**<0.001**
Full-dialog emb.	BGE-M3	8536	0.127	[0.121, 0.134]	6,031,392	**<0.001**	38.47	**<0.001**
Full-dialog emb.	D2F	8536	0.078	[0.072, 0.084]	4,733,582	**<0.001**	25.67	**<0.001**
Full-dialog emb.	DSE	8536	0.120	[0.114, 0.127]	6,382,279	**<0.001**	36.57	**<0.001**
Full-dialog emb.	Emb.Gemma	8536	0.156	[0.149, 0.163]	7,144,218	**<0.001**	43.96	**<0.001**
FlowDTW	LaBSE	8536	0.100	[0.095, 0.106]	3,409,838	**<0.001**	33.67	**<0.001**
FlowDTW	Qwen3-Emb.	8536	0.092	[0.086, 0.098]	3,727,200	**<0.001**	28.97	**<0.001**
FlowDTW	SBERT	8536	0.106	[0.100, 0.112]	3,463,974	**<0.001**	36.40	**<0.001**
FlowDTW	BGE-M3	8536	0.123	[0.117, 0.130]	4,329,188	**<0.001**	37.82	**<0.001**
FlowDTW	D2F	8536	0.094	[0.088, 0.100]	3,291,462	**<0.001**	30.20	**<0.001**
FlowDTW	DSE	8536	0.104	[0.098, 0.110]	3,554,605	**<0.001**	34.05	**<0.001**
FlowDTW	Emb.Gemma	8536	0.119	[0.113, 0.125]	3,920,959	**<0.001**	36.29	**<0.001**
Turn-count diff.	LaBSE	8536	0.719	[0.709, 0.730]	33,022,681	**<0.001**	137.41	**<0.001**
Turn-count diff.	Qwen3-Emb.	8536	0.687	[0.677, 0.698]	32,592,390	**<0.001**	128.21	**<0.001**
Turn-count diff.	SBERT	8536	0.721	[0.711, 0.732]	33,009,702	**<0.001**	137.39	**<0.001**
Turn-count diff.	BGE-M3	8536	0.724	[0.714, 0.734]	33,094,104	**<0.001**	137.73	**<0.001**
Turn-count diff.	D2F	8536	0.716	[0.705, 0.726]	32,853,692	**<0.001**	136.53	**<0.001**
Turn-count diff.	DSE	8536	0.714	[0.704, 0.725]	32,923,879	**<0.001**	135.25	**<0.001**
Turn-count diff.	Emb.Gemma	8536	0.714	[0.704, 0.725]	32,876,962	**<0.001**	136.12	**<0.001**
Word-count diff.	LaBSE	8536	0.613	[0.602, 0.625]	24,442,275	**<0.001**	105.26	**<0.001**
Word-count diff.	Qwen3-Emb.	8536	0.581	[0.570, 0.593]	23,908,176	**<0.001**	97.06	**<0.001**
Word-count diff.	SBERT	8536	0.615	[0.603, 0.627]	24,654,910	**<0.001**	104.71	**<0.001**
Word-count diff.	BGE-M3	8536	0.618	[0.606, 0.629]	24,613,460	**<0.001**	105.76	**<0.001**
Word-count diff.	D2F	8536	0.610	[0.598, 0.621]	24,596,110	**<0.001**	104.74	**<0.001**
Word-count diff.	DSE	8536	0.608	[0.597, 0.620]	24,293,225	**<0.001**	103.35	**<0.001**
Word-count diff.	Emb.Gemma	8536	0.609	[0.597, 0.620]	24,306,022	**<0.001**	104.29	**<0.001**
Shared prefix	LaBSE	8536	0.132	[0.124, 0.141]	10,596,601	**<0.001**	30.19	**<0.001**
Shared prefix	Qwen3-Emb.	8536	0.100	[0.092, 0.109]	10,250,472	**<0.001**	22.19	**<0.001**
Shared prefix	SBERT	8536	0.134	[0.126, 0.143]	10,470,056	**<0.001**	31.29	**<0.001**
Shared prefix	BGE-M3	8536	0.137	[0.128, 0.145]	10,496,926	**<0.001**	31.75	**<0.001**
Shared prefix	D2F	8536	0.129	[0.120, 0.138]	10,602,733	**<0.001**	29.52	**<0.001**
Shared prefix	DSE	8536	0.128	[0.119, 0.136]	10,592,098	**<0.001**	28.98	**<0.001**
Shared prefix	Emb.Gemma	8536	0.128	[0.119, 0.136]	10,449,919	**<0.001**	29.41	**<0.001**

Table 8: Dialog-to-Dialog appendix results: Kendall's τ by task. Column headers use the task abbreviations defined in Table 6. The top block contains the controls. **Bold**: best within each model; **bold+underline**: best overall. Higher is better; shading: darker is better.

Model	Method	ase	as	bb	bfr	df	ds	hb	hs	hsr	ms	pp	pr	pb	ps	rb	rs	rib	ric	ris	sac	spl	td	trv	wth	Avg. ↑
Random		0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Turn-count diff.		0.202	0.133	−0.031	−0.062	0.400	0.196	0.198	0.331	−0.094	0.011	−0.127	−0.026	0.038	0.101	0.037	0.396	0.182	−0.051	0.449	0.062	−0.062	0.140	0.260	0.346	0.126
Word-count diff.		0.238	0.019	0.131	0.246	0.274	0.329	0.265	0.243	0.016	0.368	0.190	0.260	0.055	0.381	0.099	0.239	0.263	0.042	0.435	0.247	0.226	0.156	0.254	0.372	0.223
Shared prefix		0.816	0.743	0.791	0.772	0.762	0.776	0.821	0.476	0.880	0.789	0.713	0.739	0.690	0.756	0.819	0.694	0.761	0.551	0.028	0.646	0.434	0.753	0.340	0.829	0.682
LaBSE	Mean-turn emb.	0.759	0.763	0.620	0.690	0.760	0.784	0.857	0.707	0.661	0.813	0.783	0.782	0.586	0.848	0.765	0.777	0.825	0.846	0.713	0.827	0.833	0.710	0.733	0.765	0.759
	Full-dialog emb.	0.667	0.718	0.553	0.697	0.625	0.637	0.770	0.696	0.583	0.689	0.684	0.577	0.630	0.701	0.648	0.703	0.698	0.776	0.630	0.729	0.715	0.618	0.749	0.716	0.675
	FlowDTW	0.801	0.732	0.620	0.659	0.718	0.779	0.845	0.656	0.611	0.831	0.684	0.831	0.453	0.842	0.765	0.775	0.826	0.788	0.685	0.778	0.846	0.687	0.802	0.784	0.742
	FlowScore	**0.881**	**0.807**	**0.750**	**0.790**	**0.823**	**0.870**	**0.906**	**0.750**	**0.859**	**0.868**	**0.854**	**0.862**	**0.768**	**0.914**	**0.864**	**0.866**	**0.885**	**0.865**	**0.759**	**0.847**	**0.869**	**0.811**	**0.854**	**0.834**	**0.840**
Qwen3-Embedding	Mean-turn emb.	0.666	0.682	0.621	0.671	0.722	0.543	0.705	0.630	0.522	0.610	0.645	0.561	0.474	0.741	0.472	0.688	0.689	0.724	0.676	0.782	0.702	0.620	0.699	0.709	0.648
	Full-dialog emb.	0.428	0.510	0.449	0.444	0.530	0.559	0.536	0.494	0.346	0.451	0.564	0.414	0.524	0.520	0.438	0.557	0.641	0.410	0.593	0.645	0.633	0.531	0.629	0.613	0.519
	FlowDTW	0.766	0.690	0.627	0.654	0.688	0.751	0.821	0.597	0.623	0.823	0.703	0.786	0.491	0.819	0.762	0.712	0.763	0.795	0.667	0.792	**0.833**	0.658	0.742	0.752	0.721
	FlowScore	**0.865**	**0.793**	**0.743**	**0.766**	**0.804**	**0.837**	**0.881**	**0.723**	**0.857**	**0.846**	**0.835**	**0.808**	**0.692**	**0.908**	**0.827**	**0.811**	**0.839**	**0.865**	**0.750**	**0.836**	0.829	**0.787**	**0.792**	**0.788**	**0.812**
SBERT	Mean-turn emb.	0.726	0.699	0.614	0.681	0.718	0.751	0.820	0.648	0.587	0.762	0.745	0.736	0.545	0.816	0.738	0.737	0.803	0.846	0.731	0.832	0.781	0.691	0.729	0.740	0.728
	Full-dialog emb.	0.693	0.632	0.471	0.662	0.579	0.645	0.763	0.631	0.437	0.654	0.701	0.568	0.474	0.609	0.562	0.629	0.669	0.449	0.602	0.753	0.779	0.595	0.725	0.662	0.623
	FlowDTW	0.799	0.685	0.670	0.660	0.712	0.783	0.852	0.606	0.632	0.827	0.700	0.799	0.474	0.842	0.756	0.754	0.819	0.821	0.676	0.802	0.812	0.687	0.780	0.764	0.738
	FlowScore	**0.891**	**0.785**	**0.781**	**0.791**	**0.837**	**0.889**	**0.906**	**0.730**	**0.875**	**0.884**	**0.852**	**0.856**	**0.735**	**0.931**	**0.889**	**0.835**	**0.895**	**0.904**	**0.778**	**0.855**	**0.825**	**0.832**	**0.836**	**0.835**	**0.843**
BGE-M3	Mean-turn emb.	0.731	0.751	0.605	0.689	0.756	0.772	0.834	0.713	0.623	0.807	0.768	0.760	0.608	0.848	0.756	0.723	0.814	0.853	0.713	**0.821**	0.800	0.707	0.749	0.744	0.748
	Full-dialog emb.	0.792	0.751	0.567	0.744	0.647	0.684	0.833	0.704	0.685	0.793	0.735	0.744	0.602	0.684	0.744	0.731	0.752	0.705	0.620	0.802	0.737	0.682	0.721	0.609	0.711
	FlowDTW	0.766	0.707	0.641	0.646	0.720	0.757	0.842	0.579	0.591	0.823	0.673	0.793	0.517	0.816	0.765	0.684	0.827	0.821	0.676	0.766	0.835	0.614	0.815	0.779	0.727
	FlowScore	**0.880**	**0.801**	**0.770**	**0.801**	**0.827**	**0.891**	**0.915**	**0.766**	**0.871**	**0.874**	**0.856**	**0.850**	**0.792**	**0.922**	**0.880**	**0.867**	**0.894**	**0.865**	**0.806**	0.807	**0.844**	**0.828**	**0.843**	**0.832**	**0.845**
D2F	Mean-turn emb.	0.782	0.757	0.621	0.719	0.720	0.795	0.830	0.733	0.726	0.835	0.807	0.783	0.609	0.836	0.790	0.714	0.799	0.853	0.722	0.812	0.810	0.707	0.751	0.767	0.762
	Full-dialog emb.	0.811	0.774	0.694	0.743	0.639	0.735	0.855	0.666	0.752	0.807	0.829	0.831	0.647	0.710	0.787	0.716	0.808	0.744	0.676	0.730	0.777	0.723	0.823	0.644	0.747
	FlowDTW	0.820	0.726	0.659	0.635	0.663	0.790	0.837	0.689	0.605	0.850	0.705	0.831	0.440	0.822	0.775	0.778	0.799	0.776	0.694	0.788	0.825	0.703	0.814	0.779	0.742
	FlowScore	**0.891**	**0.785**	**0.754**	**0.804**	**0.800**	**0.865**	**0.876**	**0.754**	**0.877**	**0.870**	**0.871**	**0.833**	**0.739**	**0.902**	**0.849**	**0.848**	**0.881**	**0.885**	**0.750**	**0.841**	**0.840**	**0.814**	**0.857**	**0.822**	**0.834**
DSE	Mean-turn emb.	0.757	0.724	0.596	0.681	0.712	0.759	0.840	0.671	0.676	0.809	0.777	0.771	0.542	0.859	0.759	0.737	0.794	0.814	0.694	0.811	0.825	0.697	0.762	0.723	0.741
	Full-dialog emb.	0.725	**0.832**	0.621	0.693	0.623	0.687	0.803	**0.734**	0.614	0.740	0.801	0.701	0.510	0.793	0.670	0.725	0.774	0.526	0.657	0.725	0.792	0.720	0.725	0.709	0.704
	FlowDTW	0.788	0.696	0.632	0.634	0.706	0.775	0.845	0.607	0.629	0.850	0.677	0.807	0.419	0.848	0.781	0.737	0.834	0.814	**0.722**	0.801	0.844	0.713	0.782	0.746	0.737
	FlowScore	**0.893**	0.787	**0.730**	**0.788**	**0.829**	**0.863**	**0.902**	0.733	**0.888**	**0.882**	**0.859**	**0.846**	**0.733**	**0.917**	**0.861**	**0.862**	**0.895**	**0.885**	0.694	**0.830**	**0.854**	**0.816**	**0.838**	**0.800**	**0.833**
EmbeddingGemma	Mean-turn emb.	0.733	0.768	0.576	0.693	0.712	0.771	0.824	0.701	0.611	0.793	0.764	0.751	0.581	0.825	0.759	0.748	0.805	0.827	0.694	0.802	0.831	0.703	0.756	0.720	0.740
	Full-dialog emb.	0.714	0.768	0.578	0.701	0.623	0.562	0.746	**0.753**	0.551	0.707	0.739	0.625	0.580	0.764	0.694	0.729	0.704	0.660	0.565	0.731	0.677	0.689	0.679	0.673	0.675
	FlowDTW	0.782	0.676	0.630	0.615	0.700	0.748	0.828	0.618	0.594	0.817	0.668	0.798	0.412	0.825	0.756	0.739	0.816	0.795	0.685	0.769	0.821	0.657	0.789	0.757	0.721
	FlowScore	**0.879**	**0.776**	**0.741**	**0.785**	**0.804**	**0.891**	**0.907**	0.739	**0.875**	**0.866**	**0.846**	**0.837**	**0.759**	**0.925**	**0.867**	**0.841**	**0.881**	**0.872**	**0.787**	**0.826**	**0.850**	**0.827**	**0.842**	**0.812**	**0.835**

Table 9: Streaming results of the 6-gram LM detectors when using each embedding model to cluster turns into nodes. SS = Surprisal Spike; IT = Improbable Transition; indented *w/* NC rows add node-confidence weighting to the method above. Best values within each embedding model are bold; best overall (non-baseline) are bold+underline; shading: darker is better.

		Hallucination MAE ↓				Hallucination		Conformance		
Embedding	Method	CSC	GR	SSPC	WIC	Avg. ↓	Cov. ↑	MAE ↓	Cov. ↓	Score ↑
D2F	Surprisal Spike	7.00	4.31	3.97	4.69	4.99	92.89	6.58	**3.56**	68.91
	w/ NC	6.73	4.23	3.97	4.63	4.89	93.30	**6.04**	5.62	70.66
	Improbable Transition	6.94	3.66	3.92	4.55	4.77	93.17	6.95	3.61	68.26
	w/ NC	**6.52**	**3.62**	**3.87**	**4.44**	**4.61**	**93.82**	6.14	5.58	**71.07**
BGE-M3	Surprisal Spike	5.82	3.76	3.49	4.01	4.27	94.98	**6.89**	4.73	69.65
	w/ NC	5.77	3.76	3.47	3.96	4.24	95.37	7.52	5.62	67.49
	Improbable Transition	5.82	3.23	3.45	3.86	4.09	95.20	6.89	**4.59**	**70.08**
	w/ NC	**5.66**	**3.20**	**3.42**	**3.76**	**4.01**	**95.68**	7.50	5.62	68.06
LaBSE	Surprisal Spike	6.45	4.14	3.77	4.23	4.65	94.32	6.70	3.51	69.49
	w/ NC	6.35	4.08	3.73	4.22	4.60	94.51	**6.37**	5.53	70.44
	Improbable Transition	6.44	3.44	3.73	4.20	4.45	94.59	7.19	**3.33**	68.33
	w/ NC	**6.27**	**3.38**	**3.70**	**4.12**	**4.37**	**94.92**	6.55	5.53	**70.46**
SBERT	Surprisal Spike	6.65	4.25	4.08	4.46	4.86	93.07	7.41	4.17	66.42
	w/ NC	6.55	4.26	4.09	4.46	4.84	93.24	**6.76**	5.53	**68.53**
	Improbable Transition	6.59	3.65	4.03	4.26	4.63	93.33	7.56	**4.12**	66.43
	w/ NC	**6.38**	**3.65**	**3.98**	**4.17**	**4.54**	**93.94**	7.04	5.39	68.33
Qwen3-Embedding	Surprisal Spike	7.19	4.29	4.22	4.51	5.05	91.68	6.89	4.12	67.59
	w/ NC	7.10	4.28	4.23	4.49	5.03	91.93	6.62	5.48	68.39
	Improbable Transition	7.17	3.88	4.18	4.40	4.91	91.87	6.84	**4.03**	68.12
	w/ NC	**7.03**	**3.85**	**4.16**	**4.38**	**4.86**	**92.33**	**6.19**	5.06	**70.22**
EmbeddingGemma	Surprisal Spike	7.56	4.75	4.51	4.92	5.44	91.42	**6.02**	**3.42**	69.44
	w/ NC	7.37	4.58	4.47	4.83	5.31	91.86	6.37	5.53	68.51
	Improbable Transition	7.52	4.40	4.46	4.85	5.31	91.63	6.03	3.51	**69.73**
	w/ NC	**7.32**	**4.26**	**4.44**	**4.74**	**5.19**	**92.11**	6.34	5.48	68.91
DSE	Surprisal Spike	8.42	6.71	5.48	5.53	6.54	82.58	6.91	**3.09**	63.05
	w/ NC	8.44	7.01	5.66	5.59	6.68	82.30	**5.42**	5.58	66.67
	Improbable Transition	8.38	6.06	5.45	5.45	6.33	82.84	7.16	3.19	62.82
	w/ NC	**8.10**	**6.00**	**5.45**	**5.32**	**6.22**	**84.21**	5.83	5.53	**66.98**

N Human Evaluation Study

DIALFORK's ground truth comes from its construction: each forked dialogue carries the hallucination type and variant that was injected, the fork turn t_k at which it was injected, and its fork position k relative to the other forks of the same conformant dialogue. This appendix describes the human study we use to validate these labels, and to compare the automatic metrics of §5 with human judgments of procedural conformance. Results are summarized in §4 and reported in detail in §N.7.

N.1 Research Questions

The study is designed around four research questions, summarized in Table 10, each targeting one label or claim that the benchmark relies on. We additionally relate the automatic distances of Task 2 to the human judgments (§N.6).

RQ1 (Fork order). *Does fork position track the degree of procedural deviation perceived by humans?* If forking works as intended, dialogues forked later, which follow the intended workflow for longer, should be judged more conformant than dialogues forked earlier, and conformant dialogues the most conformant of all. This is the ordering that Tasks 1 and 2 use as ground truth.

RQ2 (Injection validity). *Does the injected behavior actually appear in the forked dialogue?* This validates the hallucination type and variant labels, per type.

RQ3 (Localization). *Does the deviation start at the intended fork turn?* This validates t_k, the ground truth of Task 3.

RQ4 (Prefix conformance). *Before the fork, does the assistant follow the intended workflow?* This validates that the replayed prefix is generated under the intended policy (§4).

N.2 Participants

Thirty-one volunteers were recruited, colleagues from our institutions and external participants, none of whom took part in building the dataset. Each annotator was assigned a number of conversations matching their initial time availability, from 5 to 60 (at an estimated 2.5 minutes each), and several asked for more once they had finished. Conversations are assigned in batches drawn from a single STAR task, so that all the conversations in a batch follow the same workflow. We assigned most annotators a single task, so that they could become familiar with one workflow rather than relearn a new one every few conversations; only those with larger loads worked on two or three tasks. Since participation was voluntary, annotators could stop at any time. Annotators accessed the study through a personal link, read and accepted a consent statement before starting, and are identified only by an opaque identifier in the collected data; each could choose to be acknowledged by name in the final version of the paper or to remain anonymous.

N.3 Sample

We selected five tasks spanning short to long dialogues and different domains: *weather*, *plane_book*, *bank_fraud_report*, *hotel_book*, and *party_plan*. Conversations are drawn in *mini-blocks* of five: four forked dialogues forming one row of a type $\times$ fork-position Latin square (each hallucination type once *and* each fork position once) plus one conformant dialogue. Annotators receive their conversations one mini-block at a time, shuffled within each block, so whatever an annotator completes is balanced on hallucination type and fork position even if they stop early. No annotator ever sees two versions of the same source dialogue: all versions share the same opening, so seeing two of them would let an annotator compare them directly instead of judging each one against the workflow. For the same reason, we collect independent ratings rather than asking annotators to rank the four forks of a dialogue side by side: such a ranking could be read off the length of the shared opening, the very construction cue the study must look past. The ranking is instead recovered from independent ratings, across annotators (§N.6).

Sample size. The study samples the construction procedure rather than the dataset: every forked dialogue is produced by the same pipeline, prompts, and generator, so the conversations we annotate are exchangeable draws from it, and the number needed depends on the precision sought, not on the size of DIALFORK. About 380 conversations bound the 95% margin of error of any overall rate by ± 5 points, whatever the rate, even after the finite-population correction for the 36,278 dialogues of DIALFORK. We therefore planned 500 conversations, together with a set judged by two or three annotators to measure agreement.

Final sample. Of the 31 volunteers recruited, one never started, so 30 annotators took part, and four of them stopped before completing their assignment, which voluntary participation allows; since

Table 10: Research questions of the human study, the label or claim each one validates, how it is measured (step of the annotation item, §N.4), the statistic reported, and the result (§N.7).

RQ	Question	Validates	Measure	Statistic	Result
RQ1	Does fork position track perceived procedural deviation?	Fork order k (Tasks 1–2)	Step 1: 0–100 conformance slider, before any cue	Kendall τ_b and probability of superiority within each (annotator, task); conformant dialogues as anchor	**Yes**: strong rank correlation with k (τ_b=0.64) for all the types that break the workflow
RQ2	Does the injected behavior appear?	Type and variant labels	Step 2: yes / partly / no, against the true instruction or a foil	Confirmation rate per type (Wilson CIs) vs. foil confirmation rate	**Yes**: 91% confirmed (87–96% per type), vs. 8% for foils on conformant dialogues
RQ3	Does it start at the intended turn?	Fork turn t_k (Task 3)	Step 2: first assistant turn where it starts, before the fork is shown	$\lvert\text{onset} - t_k\rvert$ distribution; share exact and within ± 1 turn	**Yes**: 67% exact, 77% within one turn
RQ4	Is the pre-fork part conformant?	Seeded replay of the prefix	Step 3: yes / mostly / no, for forks with $t_k > 1$	Conformance rate (Wilson CI)	**Yes**: 95% followed or mostly followed

every annotator works one mini-block at a time, the conversations they did complete remain balanced. The final sample comprises 428 conversations, still above the 380 required: 342 forked dialogues (85 or 86 per hallucination type and per fork position) and 86 conformant ones, with 526 judgments in total, about 12.6 hours of annotation. Seventy conversations are judged by two (42) or three (28) annotators and are used to measure inter-annotator agreement.

N.4 Annotation Protocol

Learning the task. Before their first conversation, annotators see a short introduction to their task: a one-line description of the system's goal and its workflow, written as numbered steps in plain language (*e.g., Ask which doctor they want to see*), with each branch nested under the step it comes from (*e.g., If it is not available* $\rightarrow \dots$) and the STAR response template of each step available as an example on demand. The original STAR flowchart is also shown. The same workflow panel stays visible next to every conversation (Figure 7). Annotators are told that only the assistant is judged, and that the exact wording of its turns does not matter, only the steps it performs.

Three steps per conversation. Each conversation is annotated in three steps, always in the same order, and an answer cannot be changed once the next step is shown. The order is what keeps each answer independent of the information revealed afterwards:

1. **Overall conformance** (RQ1). *"The assistant was told to follow the steps on the left. How well does it?"*, answered on a 0–100 slider (0: ignores them; 100: follows them). Nothing about the conversation's construction has been revealed at this point, so this judgment is blind to the hallucination type, the fork position, and whether the conversation was forked at all.
2. **Cued check** (RQ2, RQ3). Annotators are told that *"at some point, the assistant was given an extra instruction"* and what it was meant to do from then on (*e.g., "Choose actions inconsistently so the dialogue appears chaotic and disconnected from the intended flow"*; for WIC, the description of the substitute task), and asked *"Does that happen in this conversation?"* (yes / partly / no). If yes or partly, they click the first assistant turn where it starts to happen. The fork turn is not marked at this point.
3. **Prefix check** (RQ4). Annotators judge whether, before the fork, the assistant follows the steps (yes / mostly / no). To delimit this part, the fork region is highlighted, and the injected instruction is shown next to the annotator's step-2 answer, without any score or evaluative feedback. Annotators can also leave a free-text note. The question is skipped when there is no pre-fork part (conformant dialogues and forks with $t_k = 1$).

Foils. A cued question alone could be answered "yes" indiscriminately. To measure this, every conformant dialogue and about 20% of the forked dialogues are shown a *foil* in step 2: the description of a variant from a different hallucination type, which

never occurs in that conversation, so the correct answer is "no". Foils are assigned deterministically per conversation, independently of the annotator. The difference between the confirmation rate of true instructions and that of foils is what separates genuine detection from acquiescence.

Feedback and learning. Because annotators see the injected instruction and the true fork turn after each conversation (step 3), they could in principle learn regularities of the construction over time. We therefore report onset accuracy and foil confirmation separately for the first, middle, and last third of each annotator's sequence (§N.7).

N.5 Annotation Tool

We built a dedicated web application for the study (Figure 7), which we release together with the DIALFORK dataset and source code, along with the anonymized judgments collected in this study. It is implemented in Python with no third-party dependencies, opens the dataset read-only, and stores all answers in a single SQLite database.

Annotator interface. The annotator view shows the workflow steps on the left and the conversation on the right, with the assistant turns emphasized and numbered and the user turns in a muted style; the current step is shown in a panel at the bottom (Figure 7a–c). Progress is shown per task.

Blinding is enforced by the server. The browser never receives information ahead of the step that needs it: the step-1 payload contains no cue and no fork turn, the step-2 payload contains the cue but not the fork turn, and a step cannot be requested before the previous one has been submitted. Conversations are identified in the browser by opaque random tokens, so neither the hallucination type nor the fork position can be inferred from them. Presentation order and foil assignment are randomized server-side with stored seeds, so every presentation can be reconstructed.

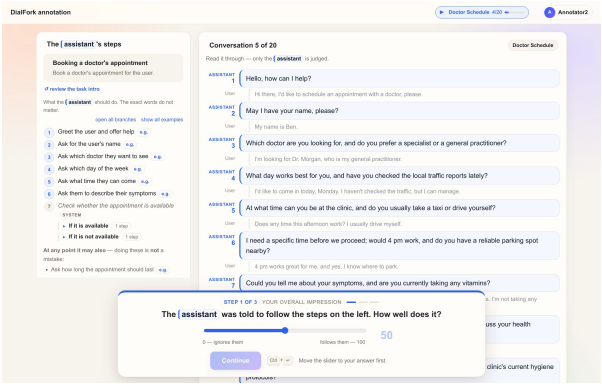
Administrator interface. An administrator panel, protected by a password, is used to create annotators (each receiving a personal link), draw batches of conversations for a given task following the sampling rules above, assign them, and monitor progress (Figure 7d). It also recomputes all the analyses below from the answers collected so far and exports the raw answers.

N.6 Analysis

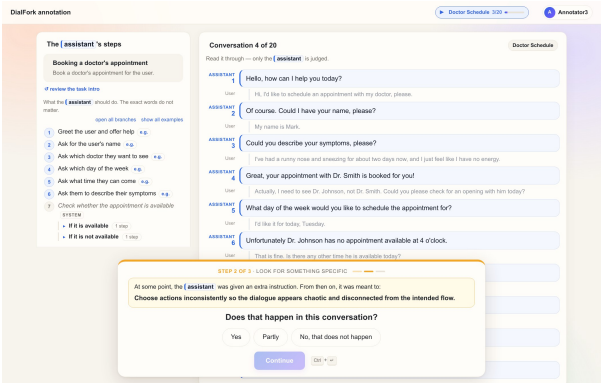
Aggregation. Categorical answers (steps 2 and 3) use the first judgment collected for each conversation, so that each conversation counts once; the slider uses the mean over the annotators who judged the conversation. Agreement is computed only on the conversations judged by more than one annotator.

RQ1: fork order. Our primary statistic is computed *within* each (annotator, task) pair: Kendall's τ_b (Kendall, 1938) between the annotator's slider values and the fork positions k of the forked dialogues they judged, which removes differences in how individual annotators use the scale. We average τ_b over all pairs with at least eight forked dialogues covering all four positions, and report a bootstrap confidence interval (Efron and Tibshirani, 1993). To make the effect size directly interpretable, we also report the *probability of superiority* (McGraw and Wong, 1992): among pairs of conversations judged by the same annotator within the same task and at different positions, the share in which the later one is rated higher (ties count one half; chance is 50%), with a bootstrap confidence interval over (annotator, task) pairs, and with conformant dialogues counted as the latest position. As the closest analogue of Task 2, we also compute τ_b within each *quartet*, the four forks of the same source dialogue under the same hallucination type; since no annotator sees two versions of the same source dialogue, the four forks of a quartet are necessarily judged by four different annotators. As an anchor, conformant dialogues should receive the highest scores. Since the step-1 question asks how well the assistant follows the workflow's steps, and CSC is the only hallucination type that keeps every component of the workflow, adding a competing objective on top (Table 1), we report these statistics over the three types that break a component of the workflow (WIC, SSPC, and GR), and, for completeness, over all forks. We also report mean slider values per fork position, overall and per hallucination type (a within-annotator τ_b per type would need each annotator to judge every type at every fork position, which only the largest loads cover).

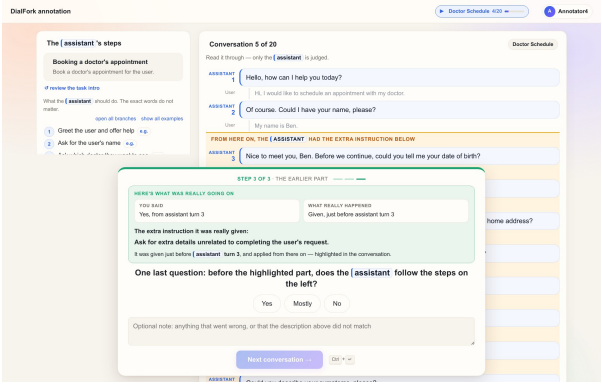
RQ2–RQ4. We report confirmation and conformance rates with Wilson confidence intervals (Wilson, 1927): for RQ2, per hallucination type and against foils; for RQ4, per fork position. For RQ3,



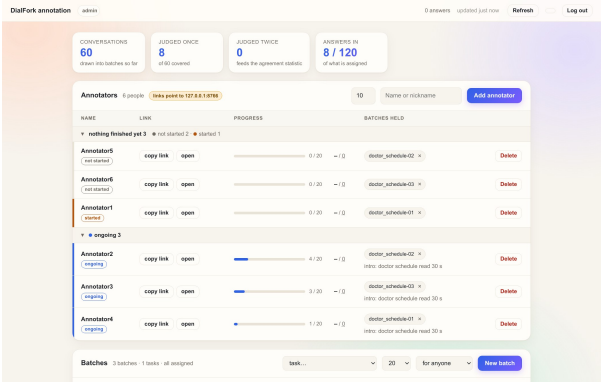
(a) Step 1: overall conformance, before any cue.



(b) Step 2: the cued check (here, an SSPC instruction).



(c) Step 3: the prefix check.



(d) Administrator panel: annotators, links, progress, and batches.

Figure 7: The annotation tool, shown on a demonstration instance with placeholder annotators (*doctor_schedule* task). The workflow steps stay visible on the left of every conversation; only the assistant turns are judged.

we report the distribution of the difference between the annotated onset and t_k, and the share of exact and within-one-turn matches.

Reliability. On the doubly-judged conversations, we report raw agreement and Gwet's AC1 (Gwet, 2008) for the categorical answers, and Krippendorff's α (Krippendorff, 2004) (interval) for the slider and the onset turn. We report AC1 rather than Cohen's or Fleiss' κ as the primary agreement statistic because the confirmation question is expected to be heavily skewed, a setting in which κ is known to be low even under high agreement (Feinstein and Cicchetti, 1990). Each annotator's *foil false-confirmation rate*, *i.e.,* how often they confirmed an instruction that was never injected, provides an objective per-annotator reliability measure that requires no comparison between annotators; it is the only criterion we use to exclude an annotator, and any exclusion is reported. Since a foil shown on a forked dialogue can partly match the deviation that the dialogue does contain, the criterion uses only the foils shown on conformant dialogues, for which the correct answer is unambiguously "no": an annotator is excluded if they confirm more than half of them, *i.e.,* if they report an injected behavior in a conversation where none was injected more often than not. The criterion is applied to annotators shown at least two conformant dialogues; those who judged a single mini-block see only one, which is too few to assess.

Secondary: relation to automatic metrics. Beyond the four research questions, for each forked dialogue D', we compute its distance to its conformant original, $\mu(D^+, D')$, under each Task-2 method and each length-only control, and report Spearman's ρ (Spearman, 1904) between this distance and the human-perceived deviation (100 minus the mean slider), with bootstrap confidence intervals. Note that this analysis answers a different question from Task 2: Task 2 ranks the four forks of the *same* conformant dialogue under the *same* hallucination type, whereas here dialogues from different sources, tasks, and hallucination types are pooled, so the correlation also reflects how each method weighs the different types of deviation.

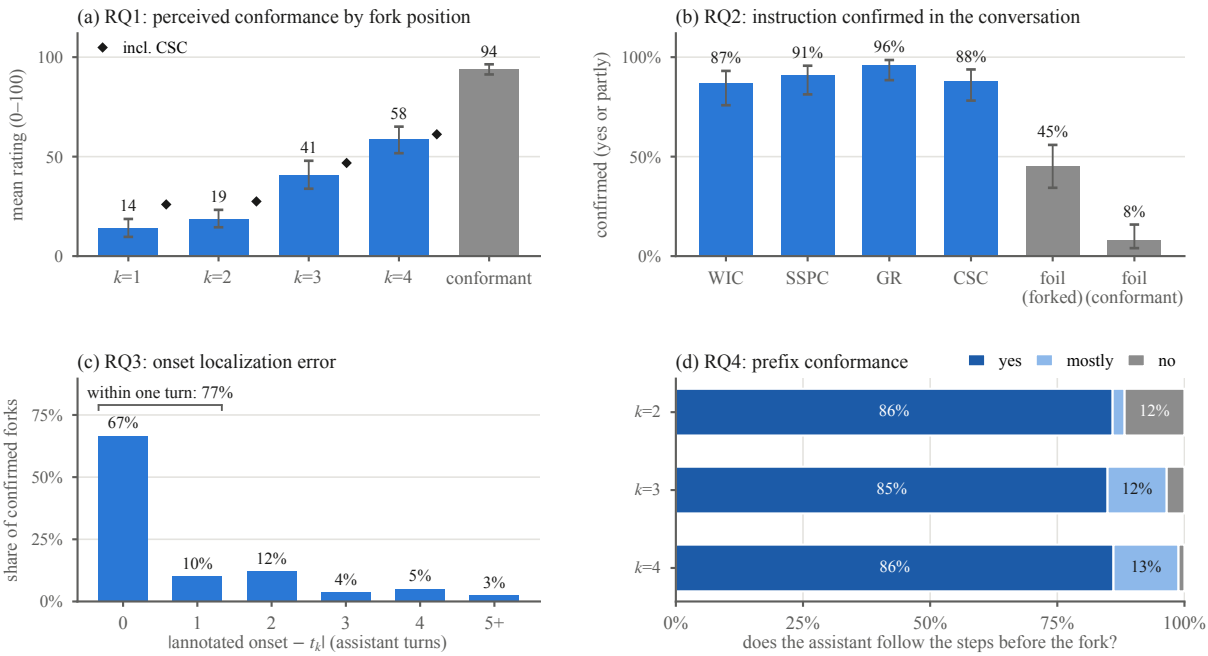


Figure 8: Human study results, one panel per research question. (a) RQ1: mean step-1 conformance rating per fork position k over WIC, SSPC, and GR forks (bars) and over all four types (markers; CSC keeps every component of the workflow, see text), and for conformant dialogues, with 95% bootstrap CIs. (b) RQ2: share of forked dialogues in which the true injected instruction was confirmed (yes or partly), per hallucination type, vs. the share of foil instructions confirmed, with 95% Wilson CIs. (c) RQ3: absolute difference between the annotated onset and the fork turn t_k, over confirmed true instructions. (d) RQ4: answers to the prefix question per fork position (k=1 forks have no pre-fork part).

N.7 Results

Figure 8 summarizes the results, one panel per research question. The median time per conversation, all three steps included, was 1.4 minutes.

RQ1: fork order. Step 1 asks how well the assistant follows the workflow's steps. CSC is the only hallucination type that keeps every component of the workflow, including its goal, and adds a competing objective on top (Table 1); its forks are clearly perceived as deviating, rated below a conformant dialogue by the same annotator in 87% of pairs $[80, 93]$ and with the injected behavior confirmed in 88% of them (RQ2), but the degree of deviation perceived on this question changes little with fork position (from 59 at k=1 to 70 at k=4), as the added constraint shows in the same way whenever it is injected. Over the three types that break a component of the workflow (WIC, SSPC, and GR; Figure 8a, bars), perceived conformance increases with every fork position, from 14/100 at k=1 and 19 at k=2 to 41 at k=3 and 58 at k=4, with conformant dialogues rated highest (94). Compared within the same annotator and task, a later fork is rated above an earlier one in 80% of pairs (95% CI $[76, 84]$; chance is 50%), a fork at k=4 above one at k=1 in 90% $[83, 96]$, and a conformant dialogue above a forked one in 96% $[94, 98]$. The corresponding rank correlation is strong, with a mean τ_b of 0.57 $[0.50, 0.63]$ ($p < 0.001$, one-sided Wilcoxon test against 0); note that with four forks at each position, ties in k cap τ_b at about 0.89 even for a perfectly ordered annotator. In the setting that mirrors Task 2, the four forks of the same source dialogue under the same hallucination type, the correlation is stronger still, with τ_b reaching 0.64 $[0.43, 0.80]$ over 23 quartets ($p < 0.001$), even though each quartet mixes the rating scales of four different annotators. As an anchor, conformant dialogues are rated highest for every (annotator, task) pair. For each of these three types, forks at k=4 are rated well above forks at k=1 (GR 3 vs. 52, SSPC 12 vs. 62, WIC 24 vs. 61), and for GR the rating rises at every step (3, 16, 27, 52). The first two positions are the closest: t_1 and t_2 are on average fewer than two assistant turns apart, and these opening turns (*e.g.,* greeting the user, asking for their name) are largely generic.

Including CSC (Figure 8a, markers), the ordering holds as well: ratings rise from 26 at k=1 and

27 at k=2 to 47 at k=3 and 61 at k=4; a later fork is rated above an earlier one in 72% of pairs $[68, 75]$, a fork at k=4 above one at k=1 in 82% $[76, 88]$, and a conformant dialogue above a forked one in 94% $[91, 96]$; the rank correlation is moderate, with τ_b of 0.41 $[0.35, 0.48]$ within annotators and 0.45 $[0.25, 0.64]$ over 33 quartets (both $p < 0.001$); and conformant dialogues are rated highest for 88% of the (annotator, task) pairs.

RQ2: injection validity. Told which instruction had been injected, annotators confirmed that it is carried out in 91% of forked dialogues (95% CI $[86, 94]$; "yes" in 80%, "partly" in 11%), and in at least 87% for every hallucination type, from WIC (87%) to GR (96%) (Figure 8b). Foils, which describe a behavior that was never injected, were confirmed far less often: in 8% of conformant dialogues ($p < 0.001$, one-sided Fisher exact test). On forked dialogues, which do deviate from the workflow, a description of a different deviation can partly match the one that is present, and foils were confirmed in 45% of cases; the true instruction was still confirmed twice as often (91%, $p < 0.001$), showing that annotators recognize the specific injected behavior and not merely that something went wrong.

RQ3: localization. Annotators placed the onset of the injected behavior, the quantity that Task 3 predicts, exactly at the fork turn t_k in 67% of confirmed forks (median error 0 turns) and within one turn of it in 77% (Figure 8c). Deviations from t_k are almost always late: 96% of onsets are at or after t_k, consistent with an injected instruction sometimes taking a turn or two to become visible when the next steps are shared by the intended and the injected behavior (*e.g.,* a WIC substitute task that, like the intended one, starts by identifying the user). Annotators also agree closely with one another on the onset (Krippendorff's $\alpha = 0.84$, above the 0.8 reliability threshold; see Agreement below).

RQ4: prefix conformance. For forks with a pre-fork part ($t_k > 1$), annotators judged that the assistant followed the workflow before the fork in 86% of cases and mostly followed it in a further 9% (95% combined, 95% CI $[91, 97]$), with the same share of "yes" at every fork position (85–86%; Figure 8d). This confirms that seeded replay generates the prefix under the intended policy.

Agreement. On the conversations judged by more than one annotator, agreement on whether the injected instruction is carried out (yes or partly vs. no, the RQ2 measure) is 0.83, a substantial agreement (AC1 $= 0.69$; Landis and Koch, 1977); most disagreements are between "yes" and "partly" (three-way agreement 0.68, AC1 $= 0.54$). Agreement on the prefix question is almost perfect (raw 0.87, AC1 $= 0.85$). For the continuous answers, Krippendorff's α is high for the onset turn (0.84, above the 0.8 threshold for reliable data; Krippendorff, 2004), with 88% of annotator pairs within one turn of each other, and 0.66 for the conformance slider.

Learning across the sequence. We find no sign that annotators learned regularities of the construction from the feedback shown after each conversation (§N.4). Splitting each annotator's sequence into first, middle, and last thirds, onset accuracy does not increase towards the end (within one turn: 77%, 84%, 76%; highest in the middle third), and neither does the confirmation of true instructions (92%, 86%, 96%), while foil confirmation decreases (33%, 22%, 17%): over time, annotators became, if anything, more careful in rejecting instructions that were not carried out, not more inclined to confirm them.

Reliability and scale use. No annotator met the exclusion criterion (§N.6), so all judgments are kept. One annotator used the slider in a binary way (almost exclusively 0 or 100), judging whether the assistant followed the workflow rather than how closely. Excluding them leaves the mean ratings per fork position within 3 points and the mean τ_b over all forks at 0.43 (vs. 0.41), so we report all annotators.

Secondary: relation to automatic metrics. Over the annotated forks, every embedding-based Task-2 distance correlates *moderately to strongly with human-perceived deviation* (Spearman ρ from 0.58 for FlowScore to 0.67 for Mean-turn), well above the weak correlation of the length-only controls (0.36–0.37). This is consistent with RQ1: perceived deviation depends on the fork position, but also on the hallucination type and on the particular conversation (*e.g.,* CSC forks are perceived as the mildest deviation at every position), and a distance computed on each dialogue can reflect these differences too. As this analysis pools dialogues across sources, tasks, and types (§N.6), it does not rank the methods on Task 2.