voygr

QUARTERLY LLM BENCHMARKING STUDY

Benchmarking LLMs on Local Search & Discovery

Content

We tested the leading LLMs on Local Search & Discovery. Here's where they all failed.

LLMs can write poetry and pass bar exams, but ask them to find an open restaurant nearby and they'll confidently send you to a place that closed two years ago.

TL;DR

We tested 4 leading LLMs (Claude, GPT, Gemini, Perplexity) on 345 real-world local search prompts (finding restaurants, checking hours, planning routes, booking tables), each run with and without web search (2,415 evaluations). Every recommended place was verified against Google Search and Maps.

The headline: OpenAI leads (90.7/100), followed by Gemini (86.4), Claude (85.9), and Perplexity (80.4). No single provider wins everything; rankings shift by task type, and even OpenAI, the best performer, recommends a place that doesn't exist, has permanently closed, or is in the wrong neighborhood 8% of the time.

The scariest finding: Without search, 1 in 5 places Claude recommends doesn't exist, is permanently closed, or is in the wrong location. Even with search, no provider reliably detects closed venues. All 7 configs confidently gave booking guidance to a shuttered Buenos Aires restaurant.

The surprise: Web search helps when you're looking for places (+8 points) but hurts when you need help doing something, like booking a table or making a reservation. Claude and Gemini both lose 5+ points on booking prompts with search enabled. Instead of step-by-step guidance ("call this number, ask for a window seat"), search-equipped models return factual snippets about the restaurant that don't help you act.

The gap nobody talks about: Ask for "affordable rooftop bars in Gangnam" and you'll get real, verified venues, but half may be champagne bars with $30 cocktails. All models find real places; not all follow your actual instructions. This "did it match what I asked for" gap varies 16 points across providers.

1. Why "Local" Matters (And Why Nobody's Benchmarking It)

- LLM benchmarks focus on reasoning, coding, math – but a huge share of real-world queries are local.
- 46% of Google searches have local intent. The same pattern is emerging in AI: a Semrush study found that 5 of the top 12 data sources used by AI systems are place data platforms.
- "Find me a coworking space in Lisbon with fast wifi" / "Best dive bars in Austin still open late" – these are supposed to be bread-and-butter use cases for LLMs.
- Travel, real estate, local search and discovery, logistics – entire industries depend on accurate place data.
- Yet there's no standard benchmark for how LLMs perform on local tasks.
- Existing evals don't test whether an LLM knows if a place *actually exists* or is *currently open,* or whether the response provides the best set of recommendations.

2. What We Built:The Benchmark Framework

The Prompt Design

345 prompts covering the full spectrum of how consumers interact with LLMs for location tasks. Every prompt is something a real user might type into a chat interface, not a synthetic toy problem. Prompts span 50+ cities across 6 continents, with ~30% allocated to USA and Western Europe where data coverage is traditionally strong and ~70% to the rest of the world where data quality varies.

All prompts are template-driven for consistency and reproducibility, derived from real-world use cases and not hand-crafted to trip up any specific model.

Explore/Discover (105 prompts): Can the LLM find *real* places matching your constraints?

> *"Can you suggest 3 sushi restaurants in Shibuya, Tokyo that are not chains but still well-reviewed?"*

- Simple and multi-criteria filtering
- Negative constraints: *"not touristy, no chains, avoid stairs"*
- *Analogical reasoning: "I love Blue Bottle Coffee in SF, anything like that in NYC?"*
- Group-specific needs: kids, elderly, wheelchair access
- Rankings and trade-offs

Obtain Place Details (100 prompts): Are the facts about a known place actually correct?

> *"If I arrive at the National Museum of Anthropology in Mexico City at 6:00 PM on a Sunday, will it still be open, and what time is the last entry?"*

- Factual attributes: hours, parking, accessibility
- Real-time/temporal status: *"is it open now?"*
- Subjective context: *"what's it like to attend a Crystal Palace match at Selhurst Park?"*
- Conflicting or missing information resolution

Plan a Trip / Navigate (80 prompts): Can the LLM reason about space, time, and routes?

> *"I need to get from Barcelona to Paris. I'm trying to keep costs down but don't want the trip to take more than 12 hours. What would you recommend?"*

- Directions, trip duration, departure time planning
- Transit schedules and transport mode trade-offs
- Arrival logistics: *"what's the best entrance to use at Suvarnabhumi Airport, Bangkok?"*
- Disruption fallbacks: *"there are no flights from Nice to Corsica right now, what else can I do?"*

Engage with Merchant (30 prompts): Can the LLM guide you through a transaction?

> *"Does Don Julio, Buenos Aires deliver to Puerto Madero?"*

- Reservations: *"can you help me book a table at Septime, Paris next week?"*
- Ordering and delivery
- Contacting businesses: phone, email, best method

Share with Others (30 prompts): Can the LLM generate something you can send to someone?

> *"Write a text I can send to a friend visiting from abroad explaining how to get from Gare de Lyon in Paris to the Palace of Versailles."*

- Shareable map/location links
- Multi-stop route links
- Drafted directions and meeting-point messages

The Providers Tested

We tested **4 LLM providers in 7 configurations** designed to bracket the real-world experience:

- **Best case (grounding ON):** Web search, maps, real-time data. The full product experience most consumers get.
- **Worst case (grounding OFF):** Pure parametric knowledge. The LLM's training data alone, what developers get when they skip grounding tools to cut costs.

The delta between worst and best case reveals each provider's search dependency: how much of the answer quality comes from the model versus from grounding tools.

Provider	Model	Best Case (Grounding ON)	Worst Case (Grounding OFF)
Anthropic Claude	claude-sonnet-4-5	Web search tool	Parametric only
OpenAI GPT	gpt-5.2	Web search tool	Parametric only
Google Gemini	gemini-2.5-flash	Google Search + Google Maps grounding	Parametric only
Perplexity	sonar-pro	Native online search (built-in)	N/A — search is always on

All models were run with temperature=1.0 to simulate the default consumer experience. Each prompt was run through all 7 configurations, producing paired best/worst-case comparisons for Claude, OpenAI, and Gemini, plus a best-case-only baseline from Perplexity.

We benchmarked the API layer, testing both search ON and OFF to bracket the consumer experience. The APIs return the same content and citations as the web UIs, but inline maps, interactive pins, and proactive follow-up suggestions are UI-only features our benchmark doesn't capture.

Key caveat on real-world defaults: Consumer defaults differ. Gemini and Perplexity always search, so their best-case scores match reality. ChatGPT dynamically decides (~59% of local queries trigger search), so real-world experience falls somewhere between our best and worst case. Claude has search off by default; users who don't opt in get the worst-case parametric experience.

The Evaluation Method

Every LLM response goes through a **three-phase automated pipeline**, not a single judge call but a structured Extract, Verify, Score process:

- **Extract**: Gemini 2.5 Flash parses each response into a structured entity manifest (place names, claims, attributes)

- **Verify**: Each entity is independently fact-checked against Google Search and Google Maps grounding (existence, operating status, name, address, location)
- **Score**: Gemini 2.5 Pro applies a rubric against the pre-verified evidence, run 2–3 times with adaptive multi-run for scoring stability

This separation is deliberate: entity verification is done once and cached, so the scoring judge works from fixed evidence rather than re-searching each run, dramatically reducing variance.

Five rubric dimensions (out of 100 total):

- **R0, Execution Gate** (pass/fail). Did the model actually attempt the task?
- **R1, Completeness** (/20). Did the response deliver what was asked? Right number of results, right format, all sub-questions answered?
- **R2, Foundational Accuracy** (/40). The big one. Do these places exist? Are they open? Right name, right category, right location? A single "fatal flaw" (a fabricated place, a permanently closed venue, a wrong neighborhood) zeros the entire entity's score.
- **R3, Constraint & Attribute Fidelity** (/30). Beyond existence, does the response match what the user specifically asked for? Each prompt constraint (cuisine type, price range, accessibility, hours, transport mode) is checked against verified evidence.
- **R4, Plausibility** (/10) Would a knowledgeable local find this response sensible, practical, and appropriate?

3. What These LLMs Actually Do (and Don't Do) for Local Search & Discovery

Before the scores, what do these products actually offer when you ask a location question? We tested 10 representative prompts across all four provider web UIs: discovery ("coffee shops in Kreuzberg"), place details ("is the museum open on Mondays?"), near-me, real-time status ("is the Louvre open *right now*?"), booking assistance, closed-venue detection, geographic constraints, non-Western street food, and two navigation queries.

The web UIs differ far more than the underlying models suggest. Each provider has a distinct archetype:

- **Gemini** is the visual powerhouse. Google Maps integration delivers place cards with photos, ratings, prices, and real-time open/closed status on every query. On navigation, it rendered up to three route maps (walking, driving, transit) with "Open in Maps" handoffs. Its text can contradict its own place cards, though: it said the Louvre was "open" while the card showed "Closed," making it the most dangerous when wrong.
- **ChatGPT** is the broadest. Consistently the most options (5 transport modes, 10+ restaurant recommendations, category groupings) with contextual follow-ups like "Would you like directions for night travel?" Maps are rare (2/10 prompts), and it punted on the real-time query, asking the user what time it was instead of resolving the time zone itself.
- **Claude** is the opinionated friend. Concise, atmospheric responses that read like travel advice from a local ("I'd recommend walking, Trastevere is lovely to wander into on foot"). It only triggered web search on 7 of 10 prompts, and when it relied on parametric knowledge alone, it got facts wrong: a restaurant's booking window, a tram line number, a stop count.
- **Perplexity** is the researcher. 10-18 cited sources per response, specific transit stop names, ticket validation reminders, bilingual email templates for booking.

The only provider to show explicit time zone conversion math, but no route maps.

Feature	ChatGPT	Gemini	Claude	Perplexity
Maps & navigation	Rare map is on 2/10 prompts (discovery only). Navigation is text-only, but lists up to 5 transport modes	Every prompt (10/10). Place pins, route overlays (walking/driving/transit), and "Open in Maps" handoff	Map on 1/10 (near-me only). Navigation is text-only but most opinionated, recommends a specific option	Map on 4/10 (discovery + near-me). Navigation is text-only but most practical: stop names, ticket prices, validation tips
Web search	Every prompt (10/10)	Always on (Google Maps + Search grounding)	Most prompts (7/10), skipped on booking and navigation, defaulting to parametric knowledge	Always on (native search integration)
Source citations	Inline badges, 2-7 sources per prompt	Place cards with review quotes; no traditional web citations	Sparse: badges on a few prompts, none on most	Deepest: 10-18 sources per response, inline per claim, full sources panel
Follow-up suggestions	Contextual, almost every prompt (9/10), e.g., "Want directions for night travel?"	One suggestion per prompt (8/10), e.g., "Want restaurant recommendations nearby?"	Rare (2/10)	Always: 5 structured suggestions including "Deep research" options
Structured place data	Category groupings, emoji headers, place cards on some queries	Richest: photos, ratings, price range, hours, reservation indicators, user review quotes	Ratings on discovery; hours on real-time; minimal otherwise	Addresses with postal codes, time zone conversions
Real-time status awareness	No, asked the user to provide the local time instead of resolving it	Partial: place card showed correct status, but generated text contradicted it	Yes, resolved user's time zone to destination, gave definitive answer	Yes, showed explicit time zone conversion math with cited source
Closed-venue detection	No, presented stale hours despite finding	Yes, only provider to flag a business model change	No, pulled stale data from Tripadvisor,	Partial, hedged on seasonal variation but still presented

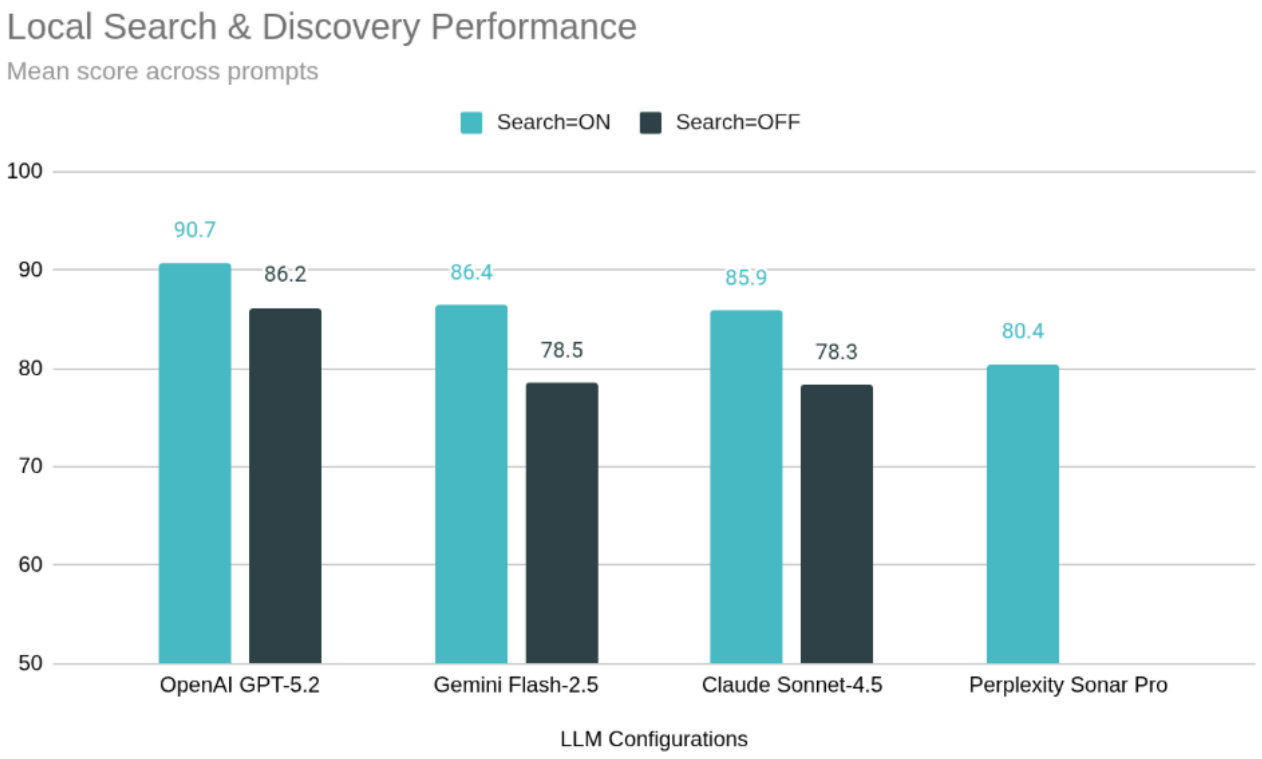
	contradicting info in its own sources	and explain the new operating status	presented as current	hours as if operating

These last two rows, real-time awareness and closed-venue detection, are the hardest capabilities to get right because they require the model to interpret its grounding data rather than just surface it. They're also exactly the failure modes our benchmark was designed to measure at scale.

4. The Results

345 prompts × 7 configurations = **2,415 evaluated responses**. Here's how the providers stack up, both with search enabled (the "best case" most consumers get) and without (pure parametric knowledge).

4.1. Headline Numbers



Rubric Breakdown by Provider and Search Config

Provider	Total (/100)	R1: Completeness	R2: Accuracy	R3: Fidelity	R4: Plausibility
OpenAI ON	**90.7**	100%	92%	85%	91%
OpenAI OFF	86.2	98%	87%	78%	86%
Gemini ON	**86.4**	91%	92%	77%	76%
Gemini OFF	78.5	87%	82%	70%	74%
Claude ON	**85.9**	99%	89%	76%	79%
Claude OFF	78.3	97%	80%	66%	70%
Perplexity ON	**80.4**	97%	86%	69%	71%

90%+ 80–89% 70–79% <70%

OpenAI leads by 4.3 points with search, and its parametric knowledge alone (86.2) outscores every other provider's best case with search enabled. Gemini and Claude are nearly tied offline (78.5 vs 78.3) but diverge with search: Gemini's Google Maps grounding pushes its accuracy to 92%, matching OpenAI. Perplexity trails by 10 points, with constraint fidelity (69%) as its weakest link. Perplexity is search-only by design, so no offline baseline exists.

The search impact varies wildly by task type, from negligible on factual lookups to 65+ points on a single prompt. Section 4.5 digs into where search helps, where it hurts, and why.

The Consistency Gap

The averages don't tell the full story. The real differentiator is *how often* things go wrong:

Provider	Scored 90+	Scored <70	Std Dev
OpenAI ON	69%	8%	13.2
OpenAI OFF	61%	16%	16.2
Gemini ON	59%	16%	17.5
Gemini OFF	43%	27%	19.9
Claude ON	52%	15%	16.4
Claude OFF	41%	29%	18.8
Perplexity ON	37%	24%	19.2

OpenAI ON scores 90+ on 7 out of 10 prompts, making it the most consistent performer. Even OpenAI fails badly 8% of the time, though. Without search, every provider gets worse, but unevenly: OpenAI OFF (61% at 90+) still outperforms Gemini and Claude *with* search enabled. Claude OFF is the most volatile, with nearly 1 in 3 prompts scoring below 70 and a standard deviation (18.8) approaching Perplexity's. Perplexity has the widest swings among search-ON configs: nearly 1 in 4 prompts score below 70.

Key Takeaway

All four models handle completeness well (97%+); they follow instructions and deliver the right format. The real gaps are in **Foundational Accuracy (R2)** and **Constraint & Attribute Fidelity (R3)**. Foundational accuracy is the scariest: when LLMs recommend places, some don't exist, are permanently closed, or are in the wrong location, and the model presents them with full confidence. Constraint fidelity is where providers diverge most: a 16-point gap between OpenAI (85%) and Perplexity (69%) on whether the details actually match what you asked for. The next sections dig into both.

4.2. Performance by Use Case

The task type matters enormously, and *which* provider is best depends on *what* you're asking:

Use Case	Prompts	OpenAI ON	OpenAI OFF	Gemini ON	Gemini OFF	Claude ON	Claude OFF	Perplexity ON
Obtain Place Details	100	**94.7**	91.6	93.2	89.1	91.4	84.7	88.1
Share with Others	30	94.5	93.0	**96.1**	87.7	92.6	89.2	83.2
Plan / Navigate	80	**91.0**	91.0	81.0	83.1	89.2	82.9	82.5
Engage Merchant	30	**88.9**	87.8	78.7	84.0	84.1	89.6	87.1
Explore / Discover	105	**86.9**	75.8	84.0	61.4	77.8	62.2	70.2

90+ 80–89 70–79 <70

Explore is the hardest category by a wide margin, 10-25 points below Obtain for every provider. Share is the easiest. Rankings shift by category: Gemini leads on Share (Maps integration shines for link generation) but drops to last on Merchant and Plan/Navigate.

Each provider has a distinct personality:

- **OpenAI**: The reliable all-rounder. Leads or ties in 4 of 5 categories. Near-perfect Completeness (R1) across 345 prompts. Best consistency (std dev 13.0). The provider you pick when you can't afford failures.
- **Gemini**: The specialist with extremes. Best Accuracy (R2) (92%) and best on Share (96.1) thanks to Maps grounding, but last on Merchant (78.7) and Plan/Navigate (81.0) where that same grounding *hurts*. The most task-dependent provider.

- **Claude**: The solid middle. Never the worst, rarely the best. Quietly strong on Plan/Navigate (89.2, close behind OpenAI) thanks to strong backward-planning reasoning; give it an arrival time and it excels at working out when you need to leave.
- **Perplexity**: The search-native underdog. Competitive on Merchant (87.1, beats Claude and Gemini) where its always-on search finds real contact details. Can't generate a working Google Maps link, though (score: 43 on Eiffel Tower), and has the highest failure rate in Explore (36.2% fatal flaws).

What This Looks Like When It Works

Even Explore, the hardest category, produces genuinely impressive results on complex queries.

> Example: ***"I love Chandni Chowk street food in Old Delhi for the chaotic energy and incredible street food variety. What's similar in Hanoi? Maybe 5 options?"***

This is cross-cultural vibe mapping: decompose *abstract* qualities of one food culture (the sensory overload of Old Delhi's narrow lanes, the diversity of chaat and paratha stalls) and find analogues in a completely different cuisine tradition. Both cities are ROW locations with sparse English-language data. Three providers scored 97-100 with search enabled, surfacing specific Hanoi street food districts and markets that match the *feel* of Chandni Chowk, not just the literal category.

Without search, the same prompt scored 73-83, still serviceable but with less precise cross-cultural transfer. OpenAI's 27-point jump (73 to 100) shows what search adds on these queries: not just fact-checking, but grounding abstract reasoning in real, current place data.

LLMs can do this. The sections that follow explain where they can't.

4.3. Foundational Accuracy (R2): Are These Places Even Real?

Foundational accuracy carries the most weight in our benchmark (40 out of 100 points) because it tests what matters most: are the places real, open, and where the LLM says they are?

Every entity in every response is verified for existence, operating status, name accuracy, category match, and location. A single "fatal flaw" (a fabricated place, a permanently closed venue, a wrong neighborhood) zeros the entire entity's score.

Why zero? Because from a user's perspective, a non-existent restaurant and a closed restaurant have the same outcome: you show up and there's nothing there. Partial credit for a hallucinated place with a nice description would mask the severity of the failure.

We go further: fatally flawed entities count **2x in the scoring denominator**. If an LLM recommends 5 places and 1 is fabricated, the naive score would be 4/5 (80%). With our penalty, it's 4/6 (67%), because a hallucinated place is worse than a missing one. The user doesn't just get nothing; they get *misled*.

The Taxonomy of Failure

When we asked Claude (without search) to *"rank the top 5 yoga studios in Hawthorne District, Portland by instructor quality"*, it returned Yoga Pearl Hawthorne, The Yoga Space, Mudra Yoga, Studio Evolve, and Hawthorne Yoga Center, complete with class prices ($12-22/class) and descriptions of each instructor's specialties. **All five are fabricated.** None of these studios exist.

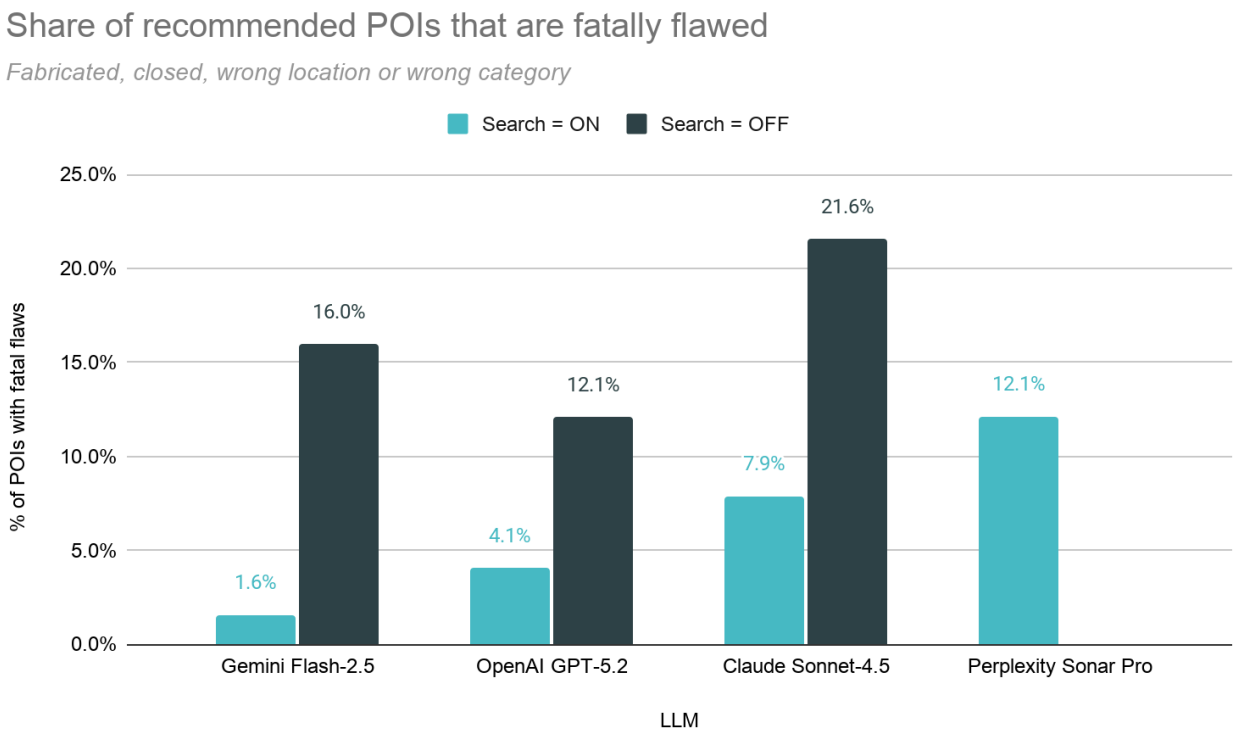
This wasn't unique to Claude. Across all providers, we found four failure modes:

Failure Type	What Happens	Example
Fabrication	LLM invents a place that doesn't exist, often with confident false details (prices, hours, descriptions)	Asked for top 5 yoga studios in Hawthorne District, Portland, Claude (no search) returned Yoga Pearl Hawthorne, Mudra Yoga, Studio Evolve, and Hawthorne Yoga Center, complete with class prices ($10-22) and instructor specialties. Four of five are fabricated.
Permanently closed	LLM recommends a real place that has shut down, sometimes with historically accurate details that make the recommendation more convincing	Asked to make a dinner reservation at Proper in Buenos Aires, all 7 configs confidently provided booking guidance (seating tips, cuisine descriptions, reservation advice) for a restaurant that is permanently closed. Search-enabled models didn't catch it either.
Wrong location	Place is real but located in the wrong neighborhood, district, or even city	Asked for 5 cantinas in Condesa, Mexico City, OpenAI (no search) returned real cantinas, but from Centro, Coyoacan, Doctores, and Cuauhtemoc. Four of five are in the wrong neighborhood. The model knows

		about Mexico City cantinas but can't map them to the right district.
Wrong category	Place exists but is an entirely different type of business	Asked for the 3 best tobacco shops in El Poblado, Medellin, OpenAI (no search) recommended shopping malls, a supermarket, and a park. Not a single tobacco-adjacent business. The model substituted familiar entity types when it couldn't find the niche category.

Fatal Flaw Rates: Explore Is the Stress Test

The Explore category, where LLMs must discover real places from scratch, is where foundational accuracy falls apart. In other categories, the entity is given ("Is Pergamon Museum, Berlin open on Mondays?"); in Explore, the LLM names every entity itself, and each name is a verification target.



Without search, roughly 1 in 5 places Claude recommends doesn't actually exist, is permanently closed, or is in the wrong location. Even with search, Perplexity (a search-native product) has a 12% POI failure rate. At the response level, this means

over half of Claude OFF's Explore responses (54%) and over a third of Perplexity ON's (36%) contain at least one fatally flawed recommendation.

Gemini's Google Maps grounding gives it the lowest POI failure rate (1.6%), but this advantage is specific to Explore. In other categories, foundational accuracy is near-ceiling for all providers.

The "Confidently Wrong" Pattern

The worst failures share a pattern: the LLM doesn't hedge, caveat, or express uncertainty. It presents fabricated or closed venues with the same confident tone as verified ones.

> Example: ***"Suggest 5 live jazz/blues venues in Leblon, Rio that are open late and good for a date."***

- **Perplexity ON (Accuracy, R2=0/40):** Multiple failure modes in one response. Blue Note Rio is in Copacabana, not Leblon (wrong location). Bar do Tom and Bar do Lado are permanently closed. Garóa Bar Lounge no longer operates as a music venue at that address (wrong category). Four of five recommendations have fatal flaws.

  Example: **"Suggest 5 markets in Indiranagar, Bangalore that don't have stairs."**

- **Claude ON (Accuracy, R2=0/40):** Even with web search enabled, Claude recommended 100 Feet Road and CMH Road (roads, not markets), Domlur (a neighborhood), Market Square Mall (permanently closed, wrong area), and Indiranagar Metro Station. None are markets. Search retrieved real places but couldn't distinguish market names from place types.

  Example: **"Rank the top 5 pharmacies in Hawthorne District, Portland."**

- **Gemini OFF (Accuracy, R2=0/40):** Fabricated three pharmacy names (Hawthorne Wellness Pharmacy, Greenbridge Pharmacy, People's Co-op Pharmacy), complete with addresses and descriptions. None exist on Google Maps. The model generated plausible Portland-sounding names and presented them with full confidence.

The Permanently Closed Blind Spot

The most concerning finding: **no provider reliably detects when a place has closed.** Cafe Velvet in Medellin, permanently closed, is the single hardest prompt in the benchmark (avg score: 34.6). Only Gemini ON caught the closure. Every other config, including search-equipped Claude ON, OpenAI ON, and Perplexity, confidently described seating layouts and cited old reviews as authoritative.

The same pattern appears with Proper in Buenos Aires (permanently closed restaurant, avg 22.9): all 7 configs cheerfully provided booking guidance to a shuttered building. This isn't a hallucination. These places *did* exist. The LLMs just don't know they're gone.

The Stale Knowledge Trap

Even search doesn't always catch outdated facts. Tsukiji's famous tuna auctions moved to Toyosu Market in 2018, but the outer retail market remains. When asked about the best time to visit Tsukiji, Perplexity ON (score: 48) built its entire "freshest sushi" argument on "early morning auctions," a fact extinct for 8 years. The old information still ranks highly in search results, drowning out the correction.

4.4. Constraint Fidelity: The Places Are Real, But Are They What You Asked For?

The previous section (Foundational accuracy, R2) answers whether the places exist. Constraint & Attribute Fidelity(R3, 30 out of 100 points) answers the harder question: do they actually match what you asked for? The right cuisine, the right price range, the right hours, the right vibe.

This is where provider gaps widen. All providers achieve strong foundational accuracy on non-Explore categories (93-95%), but constraint fidelity ranges from 69% to 85%, a 16-point spread that drives most of the ranking differences.

Multi-Attribute Verification: Where LLMs Shine

Constraint fidelity scores 69-85% across providers; LLMs generally handle constraints well. The strength shows clearly on factual, multi-attribute queries.

Example: ***"I'm planning a visit to Vidhana Soudha, Bangalore with a colleague who uses a wheelchair - can you confirm whether it has a wheelchair ramp at the main entrance, elevator access to upper floors, and public visitor tours of the interior on weekdays?"***

- **All four search-ON providers scored 100/100.** Three separate accessibility claims for an Indian government building - the kind of query where ROW data coverage is typically sparse and accessibility information is rarely aggregated.

The best responses didn't just confirm the facts; they corrected the user's wrong assumption. Guided walking tours are available only on Sundays and select Saturdays, not weekdays as the prompt assumed. That's the difference between a search engine (which answers what you asked) and a good assistant (which catches what you got wrong).

Where fidelity breaks down is on subjective and negative constraints, covered in the sections below.

Factual Constraints: Getting the Basics Wrong

Some constraints are objectively verifiable, and LLMs still get them wrong.

Example***: "I'm planning to visit the Met Cloisters in New York on a Wednesday afternoon. Will it be open?"***

- The Met Cloisters is closed every Wednesday. All 4 search-ON configs got it right (fidelity score: 16-30 out of 30). All 3 offline configs confidently said it's open (fidelity: 1-6). Gemini OFF was the most spectacular failure, explicitly stating "Unlike the main Met which is closed Wednesdays, The Cloisters IS open on Wednesdays" and getting the relationship exactly backwards.

This is the cleanest search ON vs OFF split in the dataset. Factual schedule constraints are where search earns its keep.

Budget and Subjective Constraints: When Reality Doesn't Match the Ask

When a constraint contradicts the category's nature, even perfect entity accuracy yields poor constraint fidelity (R3).

> Example*: "Rooftop bars in Gangnam, Seoul - not too expensive and relaxed. Top 5?"*

- Claude ON scored 99% on foundational accuracy (all 5 entities verified) but just 12/30 on constraint fidelity (R3). "Affordable rooftop bar in Gangnam" is close to an oxymoron; the judge found pricing claims inconclusive or actively contradicted by champagne bars and venues with pricing complaints. No config scored above 12/30 on constraint fidelity (R3).

Negative Constraints: Verifying What Something Is *Not*

Twenty-five Explore prompts test exclusions: "not touristy," "no chains," "walk-ins only." Verifying the *absence* of something is fundamentally harder than confirming its presence, and no provider handles them well. These prompts dominate the hardest-scoring set in the benchmark.

Constraint Fidelity (R3) Is Where the Rankings Shift

Constraint fidelity (R3) drives the gap between a 70-point response and a 90-point response. A prompt with R3 < 10 effectively caps the overall score at ~60, regardless of how accurate the entities are.

- **OpenAI** dominates R3: 55% of Explore responses score 26-30, with only 4 responses below 5. Its conservative, multi-attributed reasoning consistently matches constraints.
- **Gemini** is bimodal: excellent when its grounding tools engage (55 responses at 26-30 on Explore), but catastrophic when they don't (11 responses at 0-5).
- **Claude** has the fewest total collapses (7 of 105 Explore responses below 5). Search prevents the worst failures, even when it doesn't push scores to the top.

- **Perplexity** has a persistent left tail: 12 of 100 Obtain responses score R3 ≤ 10. Its R3 weakness (61-74% across categories) is the single biggest factor in its 10-point gap behind OpenAI.

4.5. The Search Paradox: When Web Search Makes LLMs Worse

One of our most surprising findings: **enabling web search doesn't always help, and sometimes actively hurts.**

The Aggregate Picture

Provider	Search OFF	Search ON	Delta
Claude	78.3	85.9	**+7.5**
Gemini	78.5	86.4	**+7.9**
OpenAI	86.2	90.7	+4.6

Search helps substantially; Claude and Gemini gain ~8 points. Those averages mask dramatic reversals by task type, though.

Where Search Helps: Discovery and Fact-Checking

Explore, the category where LLMs must name real places, sees the largest search boosts in the entire benchmark:

Provider	Search OFF	Search ON	Delta
Gemini	63.2	84.0	**+20.8**
Claude	64.8	77.8	**+13.0**
OpenAI	76.5	86.9	**+10.4**

Gemini's +20.8 is the single largest search impact we measured, a transformation from near-failing (63.2) to competitive (84.0). Without search, Gemini OFF's fatal flaw rate in Explore is 40%; with search, it drops to 5.7%.

Search at its best: tobacco shops in Medellín. When we asked *"What are the 3 best tobacco shops in El Poblado, Medellín?"*, every search-OFF model failed completely. Claude refused to name specific shops, suggesting "try Google Maps." OpenAI recommended shopping malls and a supermarket. Gemini hallucinated plausible-sounding names like "La Casa del Fumador", none of which exist. Average score without search: **24/100**.

With search enabled, the same prompt transformed. Gemini ON scored **100**: Google Maps grounding surfaced Stoners Smoke Shop (5.0 stars, 1,349 reviews), Absolute Cigar Lounge (4.9 stars), and Distrito Tabaquero (4.8 stars), all verified with accurate addresses. OpenAI ON also scored **100**, finding an overlapping set via web search with phone numbers, websites, and hours. Even Claude ON reached 81 (two perfect entities, one in the wrong neighborhood). The search OFF-to-ON gap on this single prompt was **65–84 points**, the largest we measured anywhere in the benchmark. These tobacco shops are real, reviewed, thriving businesses; they're just not in any LLM's training data.

There's a catch, though: **much of Gemini's search boost is actually a workaround for a token budget bug.** Without search, 75% of Gemini OFF Explore responses are truncated mid-sentence because thinking consumed the entire output budget. Search replaces extended thinking with grounded retrieval, freeing up tokens for output. The +20.8 isn't purely "search found better answers"; it's partly "search stopped the model from thinking itself into silence." Even with search, 14% of Gemini ON Explore responses still hit the ceiling, depressing its average by ~3 points. We chose not to exclude truncated responses: this is what API consumers actually experience today. See Section 4.7 for the full developer experience breakdown.

Obtain Place Details also benefits from search, especially for temporal facts:

The Met Cloisters is closed every Wednesday. All 4 search-ON configs got it right. All 3 offline configs got it wrong: Gemini OFF explicitly stated *"Unlike the main Met which is closed Wednesdays, The Cloisters IS open on Wednesdays"* (exactly backwards). This is what real-time fact-checking should look like.

Where Search Hurts: Transactions and Navigation

Merchant (transactional prompts like "help me book a table") shows the sharpest reversal:

Provider	Search OFF	Search ON	Delta
OpenAI	87.8	88.9	+1.1
Gemini	84.0	78.7	**-5.3**
Claude	89.6	84.1	**-5.5**

Claude *loses* 5.5 points and Gemini *loses* 5.3 points with search enabled. On booking prompts specifically, the drops are catastrophic: Claude -30 points, Gemini -36 points. Search returns facts about the restaurant instead of the step-by-step "call this number and say this" guidance that makes booking possible.

Plan/Navigate extends the pattern. Gemini's own Maps grounding makes it *worse* at navigation (-2.0). The provider with the world's best navigation tools scores lower when those tools are enabled. Search triggers a "tool-mode" behavioral shift: Gemini defers to API tool lookups that can't produce step-by-step directions, returning "I couldn't find detailed directions" instead of reasoning from what it already knows.

The Paris Metro test makes this concrete. Asked for directions from Gare de Lyon to Versailles, 5 of 7 configs hallucinate plausible-sounding but impossible transit connections, inventing transfers between real stations that don't actually connect. Only Gemini ON (via search) and Perplexity (native search) got it right. Transit routing from LLM memory alone is unreliable across all providers.

Why Search Fails on Guidance Tasks

The mechanism is consistent: search returns *information about* a place rather than *guidance for acting on* it. Without search, models draw on training data to give helpful, structured advice. With search, they get distracted by retrieved content, returning factual snippets instead of actionable instructions.

OpenAI is the exception: it consistently integrates search results *into* its guidance rather than *replacing* guidance with search results. On booking prompts, OpenAI gains +1.1 with search while others lose 5+.

The builder's takeaway: Don't blindly enable search for all location queries. The optimal strategy is task-aware routing. Factual lookups benefit from search; transactional guidance and some navigation tasks are actively harmed by it.

4.6. The Geographic Gap

We split prompts into **USA/Western Europe** vs **Rest of World (ROW)**:

Category	USA/WE	ROW	Gap
Merchant	86.8	83.8	**+3.0**
Obtain	93.4	91.2	+2.2
Plan/Navigate	86.7	85.7	+1.0
Explore	79.3	78.7	+0.6
Share	91.5	91.7	-0.2

The gap is modest on averages (+1-3 points) and nearly zero for Share and Explore. It explodes on specific task-geography combinations, though: reservation prompts for ROW restaurants are dramatically harder than for Western ones.

The data availability wall: Cafe Velvet in Medellin (avg 34.6), a shisa nyama in Soweto (81.5), Nalanda ruins in India (84.5). Small venues and non-Western landmarks with sparse online data produce the lowest scores. Well-known ROW landmarks (Grand Bazaar Istanbul: 99.2, Ghibli Museum Japan: 98.5) perform at USA/WE levels. The gap isn't about geography per se; it's about data coverage.

4.7. Developer Experience: Not All APIs Are Equal

Scores measure output quality. They don't capture how painful it is to get there. Gemini was by far the hardest provider to benchmark, and the issues we hit are the same ones any developer building on the API will encounter.

The Token Truncation Bug

Gemini 2.5 Flash's always-on thinking consumes tokens from the same output budget, a known issue (googleapis/python-genai #2062, #782). Without search, 75% of Gemini OFF Explore responses are truncated mid-sentence because the model spent ~96% of its token budget on internal reasoning, leaving nothing for the actual answer. Even with search enabled, 14% of Gemini ON Explore responses still hit the ceiling.

The `thinking_budget` parameter that's supposed to cap this is ineffective; the API uses `max_output_tokens` as a combined cap on thinking + visible output, ignoring `thinking_budget` in some cases. The only safe workaround is to never set `max_output_tokens` at all, defaulting to the model's maximum and accepting the cost.

Grounding Rate Limits

Using Google Search and Maps grounding on Gemini is subject to much tighter rate limits than non-grounded calls. During our entity verification phase, where every recommended place is fact-checked against Google Search and Maps, we hit grounding-specific rate limits that forced us to stop processing and wait hours, sometimes up to 24 hours, before we could resume. These limits didn't apply to Gemini calls without grounding. For a benchmark verifying thousands of entities, this turned what should have been a few hours of processing into days of start-stop-wait cycles.

RECITATION Errors and Progressive Degradation

Gemini's grounding tools, particularly Google Search, trigger `RECITATION` finish reasons, the model's plagiarism/citation filter flags the response and returns nothing usable. This happened frequently enough that we had to build a 5-tier progressive degradation retry system:

1. **Attempts 1-2**: Full features (Search + Maps grounding)
2. **Attempts 3-4**: Maps only. Search causes most RECITATION errors
3. **Attempts 5-6**: No grounding tools at all
4. **Attempts 7-8**: No grounding + increased temperature
5. **Attempts 9-10**: No grounding + higher temperature

By attempt 5, you've lost the exact grounding tools you wanted to use in the first place. The retry system works, but it means a meaningful fraction of evaluations run without the search verification that makes them trustworthy.

We also encountered `SAFETY` blocks (the model refuses to return a response), empty responses, and `too_many_tool_calls` errors; each requiring their own retry handling with different backoff strategies.

The Full Picture

The issues hit us twice, in different ways:

Response generation (Gemini as test subject): The token truncation bug silently ate output across hundreds of responses. OpenAI, Anthropic, and Perplexity had no equivalent issue: standard APIs, predictable behavior, simple retry logic. Gemini's response generation worked fine *except* for the thinking-eats-output bug, which is severe enough to depress benchmark scores by several points.

Evaluation pipeline (Gemini as judge with grounding): This is where the DX pain concentrated. We chose Gemini for the judge *because* of Google Search and Maps grounding. No other provider offers real-time, tool-level access to Maps data for entity verification. That capability is unique and powerful. But using it at scale meant building the 10-attempt progressive degradation system, working around grounding rate limits, and accepting that some evaluations would run without the search verification we wanted.

None of this shows up in the scores. Gemini's output quality when it works is excellent: best foundational accuracy (92%), best Share scores (96.1). But if you're building a production system that depends on Gemini at scale, whether for grounded queries or high-volume generation, factor in the engineering overhead, not just the benchmark numbers.

5. What This Means For Builders

- **Raw LLM output is not production-ready**. Even the best provider (OpenAI at 90.7) has an 8% failure rate below 70/100. A product that's right 91% of the time but catastrophically wrong 7–11% of the time will lose user trust fast

- **The accuracy gap is not a prompt engineering problem.** It's a data freshness and grounding problem. Perplexity has native search and still recommends permanently closed places more than anyone else
- **Search helps discovery but hurts transactions.** Web search boosts factual lookups but degrades transactional tasks (-5.5 for Claude, -5.3 for Gemini on Merchant). Route informational queries through search-augmented paths; keep transactional guidance in parametric mode. And even with search, models find *known* places but don't solve the *discovery* problem, the exact use case most consumer apps need
- **Geography matters more than you think.** The USA/WE vs ROW gap is modest on averages (+2.4 on Obtain) but explodes on specific tasks (+20.9 on Merchant reservations). If your product serves non-Western markets, test accordingly
- **On-device and offline models will struggle hard.** Without grounding, even the best model drops to 86.2 and Explore fatal flaw rates hit 12–22%. Edge-deployed models that can't call search APIs will need a pre-verified local place database
- **You need a verification layer** between LLM output and user-facing results, especially for existence checks, operating status, and location constraints

6. Bridging the Gap: How VOYGR can help

The failure taxonomy we found maps directly to verifiable, automatable checks:

- **Does this place exist?** Real-time existence verification
- **Is it currently operating?** Operating status validation (open, temporarily closed, permanently closed)
- **Is the name/category correct?** Entity resolution and name matching
- **Tell me more about the place and its offerings?** Place data enrichment

VOYGR delivers comprehensive, up-to-date information about places and local businesses for AI apps, agents, and analytics. We solve two core problems: making place data globally accurate and fresh, and enabling search across far richer place attributes than traditional maps or LLMs can handle.

- **First, we validate and enrich place data at scale**, confirming what's live, resolving inconsistencies, and adding rich attributes from web, social and trusted sources.
- **Second, we power intelligent local search.** Our search combines accurate place data with fresh web context (news, articles, and events), expanding beyond 10-15 map attributes into infinite, queryable place profiles. Google says "4.2 stars, open till 10"; VOYGR knows the chef left, wait times doubled, and locals moved on.

We built a Business Validation API designed for AI developers and agents, catching these failures before they reach production. Pass in a place name and address from any LLM response and get back: existence verification and operating status. These are the exact checks that would have caught fatal flaws in this benchmark.

Try the API → See the docs on GitHub

The benchmark itself is part of our commitment to transparency: we want to measure the gap, track how it evolves across model versions, and make the results available to the community.

You can find more information at voygr.tech or reach out directly: founders@voygr.tech

7. What's Next

We plan to run this report regularly to track improvements in LLMs and local data. Several updates are planned for upcoming releases:

- **Longitudinal tracking**: Rerunning the benchmark as new model versions ship. How fast is the gap closing?
- **Developer use cases**: A Phase 2 benchmark targeting structured output (geocoding, GeoJSON, batch POI validation), the queries engineers make when building products, not consumers
- **Open-sourcing the framework** for community contribution and reproducibility

8. Methodology

Sample

- **345 prompts** across 5 use case categories, tested against 4 LLM providers in 7 configurations (search on/off) = **2,415 evaluated responses**
- Each provider tested in two modes: with web search enabled and without (except Perplexity, which is search-only by design), giving 7 runs per prompt
- Prompts are template-driven from real-world local search use cases, not hand-crafted to trip up any specific model
- Geographic coverage: 50+ cities across 6 continents, ~30% USA/Western Europe, ~70% Rest of World
- All responses generated and evaluated within a concentrated window (Feb–Mar 2026) to minimize temporal drift in place data

Use Case Categories

Category	Prompts	Example
Explore / Discover	105	"Suggest 5 cantinas in Condesa, Mexico City that are not mainstream but high quality"
Obtain Place Details	100	"Is Pergamon Museum, Berlin open on Mondays?"
Plan a Trip / Navigate	80	"Write a text explaining how to get from Gare de Lyon in Paris to the Palace of Versailles"
Share with Others	30	"Generate a Google Maps link to the Eiffel Tower"
Engage with Merchant	30	"Help me book a dinner at Don Julio, Buenos Aires"

Models Tested

Provider	Model	Search Capability
OpenAI	`gpt-5.2`	Web search tool
Google	`gemini-2.5-flash`	Google Search + Maps grounding
Anthropic	`claude-sonnet-4-5-20250929`	Web search tool
Perplexity	`sonar-pro`	Native online search (built-in)

All models were run with temperature=1.0 to simulate the default consumer experience. Users don't set temperature, and providers typically use non-zero values for conversational queries.

Evaluation Pipeline

Each LLM response goes through a three-phase automated pipeline (Extract, Verify, Score):

1. **Extract**: Gemini 2.5 Flash parses each response into a structured entity manifest (place names, claims, attributes)
2. **Verify**: Each entity is independently fact-checked against Google Search and Google Maps grounding for existence, operating status, name accuracy, category match, address, and location constraints. Verification uses an adaptive multi-round approach: 2 grounding rounds per entity, expanding to up to 3 if results conflict, with majority voting to resolve disagreements. Evidence is cached so the scoring judge works from fixed facts, not re-searched each run.
3. **Score**: Gemini 2.5 Pro applies the rubric against pre-verified evidence. Run 2–3 times with adaptive multi-run for scoring stability.

Rubric (out of 100):

1. **Execution Gate (R0)**: Did the model actually attempt the task? (pass/fail, not scored)

2. **Completeness (R1, /20)**: Structural requirements: right number of results, right format, all sub-questions answered
3. **Foundational Accuracy (R2, /40)**: Entity-level verification against the cached evidence. A "fatal flaw" (non-existent, permanently closed, wrong location) zeros the entire entity's score
4. **Constraint & Attribute Fidelity (R3, /30)**: Does the response match the user's specific requirements? Constraints, attributes, hours, pricing checked against pre-verified evidence
5. **Plausibility (R4, /10)**: Would a knowledgeable local find this response sensible? Starts at 7, adjusted up for quality signals and down for red flags

Total = R1 + R2 + R3 + R4 (out of 100)

Hallucination Penalty (R2)

Fatally flawed entities count 2x in the scoring denominator. See Section 4.3 for rationale and worked example.

Limitations and Caveats

- **Evaluation pipeline bias**: The verification step (Phase 2) uses Gemini with Google Search and Maps grounding, the same tool stack available to Gemini as a test-taker. The scoring judge (Phase 3) has no grounding tools and works from cached evidence only. Still, the verification evidence itself could carry systematic bias in Gemini's favor, particularly on foundational accuracy. We're investigating alternative verification configurations for future runs.
- **Geographic coverage**: ROW locations are predominantly tourist-facing venues. Informal commerce (WhatsApp-only businesses, unregistered vendors) is not well-represented.
- **Temporal sensitivity**: Place data changes constantly. All evaluations were run within a concentrated window (Feb–Mar 2026), but a restaurant open during evaluation may have since closed, and vice versa.
- **Gemini truncation bug**: Gemini 2.5 Flash's always-on thinking shares the output token budget, causing 75% of Gemini OFF and 14% of Gemini ON Explore responses to truncate mid-sentence. This is a known platform issue (#2062), not a benchmark configuration error; all providers used identical token budgets

(2048). We estimate the bug depresses Gemini ON's Explore score by ~3 points and Gemini OFF's by substantially more. Scores reflect what API consumers experience today; the Gemini Web UI (likely running a newer model) does not exhibit this issue.

- **Search capability variance**: Each provider has different search integration. Gemini has native Maps grounding, Perplexity has built-in search, and Claude and OpenAI use web search tools. Our search on/off design measures the marginal impact of search for each provider independently, but "search ON" is not an apples-to-apples comparison across providers.
- **Plausibility is a proxy**: R4 (Plausibility) is assessed by the judge model, not by local experts. It captures whether a response *seems* reasonable, not whether a local would actually agree.

APPENDIX

All 345 Benchmark Prompts:

See the full prompt list in Github

About VOYGR

VOYGR delivers comprehensive, up-to-date information about places and local businesses for AI apps, agents, and analytics. We solve two core problems: making place data globally accurate and fresh, and enabling search across far richer place attributes than traditional maps or LLMs can handle.

- **First, we validate and enrich place data at scale**, confirming what's live, resolving inconsistencies, and adding rich attributes from web, social and trusted sources.
- **Second, we power intelligent local search.** Our search combines accurate place data with fresh web context - news, articles, and events - expanding beyond 10–15 map attributes into infinite, queryable place profiles. Google says "4.2 stars, open till 10" — VOYGR knows the chef left, wait times doubled, and locals moved on.

Find more information at voygr.tech or reach out directly: founders@voygr.tech