Calibration-Driven Demand-Aware Decoding for Structured Output: A Calibration-Based Deployment Criterion for JSON Schemas

Anonymous ACL submission

Abstract

Grammar-constrained decoders enforce JSON schemas but cannot tell whether the model dropped a field the user requested or invented one the user did not. We study *demand-aware decoding*: a one-call classifier marks each optional field, and the demand set drives either scaffold pruning (DAS) or schema promotion to `required` (CGA). Across 11 checkpoints and five JSON protocols, the same one-line edit helps some (checkpoint, protocol) cells and hurts others; we show that a single classifier-side calibration gap predicts which. The gap is informative only in a middle band of protocol demand density, computable from the schema and a small prompt sample with zero model calls. This yields a two-stage deployment rule that requires no model call at the protocol-axis stage and recovers Pareto gains on calibrated classifiers (Claude Haiku/Sonnet recover -23 to -24pp of fabrication on FHIR at near-zero coverage cost), while correctly predicting null effects on boundary protocols (OCPP, OpenAPI). A 3×3 classifier-by-generator swap and a same-set audit on the free-form prior support the mechanism.

1 Introduction

When a language model is asked to produce structured output, it must decide which *optional* schema fields to fill. A grammar-constrained decoder (Willard and Louf, 2023; Beurer-Kellner et al., 2023; Scholak et al., 2021; Dong et al., 2024; Geng et al., 2023; Gat, 2024; Norick et al., 2023) makes the result valid JSON but says nothing about which fields the user asked for. Two failures follow from this. The decoder follows the shortest valid path (Holtzman et al., 2020) and drops optional fields that the prompt explicitly requests: FHIR `birthDate` is retained in 0 of 17 birthDate-mentioning prompts across four open-weight models under constrained decoding, while every free-form baseline retains $17/17$. Schemas with unbounded numeric types let the decoder emit arbitrarily long digit literals that exhaust the token budget mid-value, producing 8–22% unterminated JSON on OpenAPI samples. The validator does not detect either failure.

We make the schema or its scaffold *demand-aware*. A short classifier prompt asks for each optional field whether the user request demands it. The positive set then drives one of two interventions. In *Conditional Grammar Augmentation* (CGA), the demanded fields are promoted from `properties` to `required` before constrained decoding. In *Demand-Aware Skeleton* (DAS), the undemanded fields are dropped from a typed scaffold that the model is asked to copy. Both are one-line edits acting on the same decision: which optional fields should the output contain. The classifier and the generator are the same checkpoint per cell, unless noted otherwise; §10 separates the two.

Main empirical finding. On a 330-sample, four-protocol main benchmark (with GraphQL as an external low-density control; §7), this one-line edit produces a signed effect that varies between (checkpoint, protocol) cells and is explained by a single *a priori*–computable calibration mechanism (§7). The mechanism predicts which cells yield Pareto outcomes (lower fabrication at preserved coverage) and which incur coverage loss; deployment requires zero model calls at the protocol-axis stage. Table 1 summarizes the signed effect on FHIR R4 (N=57). We label each (checkpoint, protocol) cell by Δ = DAS − SFG in coverage and fabrication (formal thresholds in §4):

- **Pareto**: lower fabrication *without* losing coverage — the deployment win.
- **help**: higher coverage (DAS keeps fields SFG misses); fabrication unchanged.
- **reversal**: lower coverage — DAS *strips* demanded fields; harmful.

Checkpoint	Δcov	Δfab	label
Qwen-2.5-3B	+1.8	−1.2	ambig.
Qwen-2.5-7B	+6.5	−0.8	**help**
Phi-3-mini-4k	−12.8	−3.6	**reversal**
Mistral-7B-v0.3	+0.0	−13.8	**Pareto**
Llama-3.1-8B	−27.3	−6.0	**reversal**
GPT-4o-mini	−22.2	−4.3	**reversal**
GPT-4o-full	−25.3	−9.5	**reversal**
Claude Haiku 4.5	−1.0	−24.1	**Pareto**
Claude Sonnet 4.5	−2.0	−23.3	**Pareto**
Claude Opus 4.5	+0.0	−0.9	null
Gemini 3.1 FL	+0.0	+6.0	ambig.

Table 1: Δ(DAS−SFG) on FHIR R4 (N=57, pp); labels defined just above. Robustness to oracle choice in §7.

- *mild reversal*: small coverage loss (between −10 and −5pp).
- *null*: both $|\Delta| < 5$pp; no measurable effect.
- *ambiguous*: any remaining sign combination.

The eleven checkpoints split as: **Pareto** on Mistral-7B and Claude Haiku/Sonnet 4.5; **help** on Qwen-2.5-7B; *null* on Claude Opus 4.5; **reversal** on Phi-3-mini, Llama-3.1-8B, and GPT-4o-mini/full; *ambiguous* on Qwen-2.5-3B and Gemini 3.1 FL.
The calibration regression in §7 attaches each cell to a single classifier-side observable, μ, so the above split is the predicted output of a deployment rule, not an unexplained empirical pattern.

Why not a free-form-prior effect? The simplest alternative explanation is that the families differ in how many optional fields they emit on zero-shot, so the prune lands on different populations. The four frontier checkpoints emit not just the same raw count, but exactly the same set of demanded pairs on FHIR: each of GPT-4o-mini, GPT-4o-full, Claude Haiku 4.5, and Claude Sonnet 4.5 emits the same 82 pairs out of the same 99 oracle-positive pool (pairwise Jaccard = 1.0; full set-agreement audit in §5). The free-form prior on the demanded fields is therefore not the source of the divergence.

Contributions. This paper contributes:

1. **A method.** *Demand-aware decoding* — a one-call optional-field classifier whose positive set either prunes an SFG scaffold (DAS) or promotes fields to `required` in the schema (CGA); a one-line edit on top of any constrained decoder (§3).
2. **A deployment rule.** A two-stage criterion (§7): Stage 1 estimates the protocol-level $\Pr[Y]_{\text{oracle}}$ from the schema's optional-field set, representative prompts, and the released oracle rule (zero model calls) and applies the method only when it lies in $[0.3, 0.6]$; Stage 2 picks a classifier with $|\mu_{\text{emit}}| < 0.05$ on a small calibration sample.
3. **A benchmark and an explanation.** Eleven checkpoints × five JSON protocols (§4); a 3×3 classifier-by-generator swap shows the *classifier* identity (not the generator) drives the direction (§8); a same-set audit and a Random-K decomposition rule out the free-form-prior alternative (§5, §6).

Scope. The benchmark and the calibration regression target the JSON encoding (§3). A cross-serialization audit on six alternatives (Appendix G) shows that the JSON-side direction labels do not transfer one to-one to YAML, XML, TOML, NDJSON, Python-dict or S-expressions; extending the mechanism to non-JSON pipelines is left to future work.

2 Related work

Constrained decoding. Grammar-constrained decoders mask logits against a JSON Schema or CFG so that emitted tokens stay in-grammar (Willard and Louf, 2023; Beurer-Kellner et al., 2023; Scholak et al., 2021; Dong et al., 2024; Geng et al., 2023; Gat, 2024; Norick et al., 2023); vendor-provided structured-output APIs (OpenAI, 2024a,b; Anthropic, 2024; Google DeepMind, 2024) solve structural validity but inherit the same demand-blindness — the schema, not the prompt, decides which optionals are expected. **Prompt-level scaffolding.** Skeleton-of-thought (Ning et al., 2024), plan-and-solve (Wang et al., 2023), and JSON-mode prompting (He et al., 2024) pre-commit a structural plan before the generator fills it but, again, do not condition the plan on the prompt's demanded subset. **Demand-awareness as a contribution.** Our intervention is a one-line, prompt-conditional extension: a single classifier call labels each optional Y/N, and the positive subset is either promoted into `required` (CGA) or kept in the scaffold (DAS). Field-level demand classification has been used in slot filling (Henderson et al., 2014; Louvan and Magnini, 2020) and semantic parsing (Yu et al., 2018), but coupling it with constrained decoding to produce a signed effect driven by calibration on coverage and fabrication is new.

3 Setup

Benchmark. N=330 prompts across four protocols selected to span two input-measurable axes. The *Lexical anchor density* λ of a protocol is the fraction of (sample, optional-field) pairs for which the prompt contains a trigger substring of the field's name (exact, camelCase split, or curated synonym). The *value-content overlap* ν is the fraction of pairs for which the prompt contains a token that matches the expected value class of the field (date, number, enum, URI, proper noun). Concretely:

Protocol	N	λ	ν
OCPP-J 1.6	73	0.15	0.95
FHIR R4	57	0.50	0.66
OpenAPI 3.x	100	0.78	0.49
Function-calling	100	0.58	0.58

Table 2: Per-protocol counts and the two prompt-side demand-density parameters: λ is the lexical anchor density (fraction of pairs whose prompt names the field) and ν is the value-content overlap (fraction of pairs whose prompt contains a token of the field's expected value class). Standards cited in text: OCPP (Open Charge Alliance, 2017), FHIR (HL7 International, 2019), OpenAPI (OpenAPI Initiative, 2024).

OCPP and FHIR are real-world deployed standards. The OpenAPI specification used is a synthetic Petstore-style document chosen to isolate the high-λ, low-ν corner of the plane; the function-calling benchmark is a separate, independent Petstore agent-tool collection that covers web search, mail, calendar, database, and file operations, in the spirit of recent function-calling benchmarks (Patil et al., 2024; Qin et al., 2024; Li et al., 2023; Yan et al., 2024). The two synthetic protocols are not the same document and exercise different optional-field structures (depth, mixed types, enum density); we use "Petstore-style" only as a thematic descriptor of the OpenAPI document and not of the tool collection.

The two metrics λ and ν are deterministic per-protocol functions computed before any model is called. λ counts trigger substrings of each optional field's name in the prompt (exact, camelCase, or curated synonym; the synonym list is released); ν counts tokens that match the expected value class of the field (ISO date, integer, enum value, URI scheme, or proper-noun pattern). Both metrics, the synonym list, and the trigger code path are released as part of the benchmark.

Models. Eleven checkpoints across seven families: open-weight Qwen-2.5-3B/7B (Qwen Team, 2024), Phi-3-mini-4k (Abdin et al., 2024), Mistral-7B-Instruct-v0.3, Meta-Llama-3.1-8B-Instruct (BF16, greedy decoding via HuggingFace Transformers); closed-API GPT-4o-mini, GPT-4o (OpenAI Responses API), Claude Haiku 4.5, Sonnet 4.5, Opus 4.5 (Anthropic Messages API), and Gemini 3.1 Flash-Lite (Google AI Studio). Gemma 4 31B-IT is reported as a zero-shot prior reference on FHIR. All API calls were made at temperature 0; exact version strings and dates appear in Appendix A.

Methods. We compare three methods that share an input *(prompt p, schema $\mathcal{S}$)* but differ in what the generator sees (Table 3). *SFG* (Skeleton-First Generation) is the baseline: the schema is compiled into a typed JSON-like scaffold (one «`typed placeholder`» per property, marked `optional` for non-required) and the model copies the scaffold filling only the prompt-demanded slots; no classifier call. *DAS* (Demand-Aware Skeleton) and *CGA* (Conditional Grammar Augmentation) both add *one* classifier call per sample (the same one for both; $\leq$ 64 output tokens) that marks each optional field Y/N. DAS uses the positive set D to *prune* the SFG scaffold (removing slots $f \notin D$) before the same fill step. CGA uses D to *promote* those fields from `properties` into `required` in the schema before grammar-constrained decoding (Outlines on open weights; OpenAI structured-outputs on API-only checkpoints). Algorithm 1 is the single shared pseudocode.

	Cls.?	Decoder input	Direction lever
SFG	—	scaffold w/ all optionals	baseline
DAS	yes	scaffold pruned to D	remove unmarked
CGA	yes	schema with D `required`	force marked

Table 3: SFG / DAS / CGA at a glance. All three take the same (prompt, schema); only DAS and CGA add one classifier call producing the demanded set D.

The classifier prompt and the per-protocol skeleton renderer are in Appendix E; the demand classifier's system prompt is in Appendix B; the oracle's surface-form matcher is documented in Appendix D. The classifier and generator are the same checkpoint per cell, unless explicitly noted (§8).

Metrics. We use three field-level metrics on (sample, optional-field) pairs. *Validity* is the

Algorithm 1 One pass of DAS / CGA on a sample.

Require: prompt p, schema $\mathcal{S}$, LLM $\mathcal{M}$, mode $\in \{\text{DAS}, \text{CGA}\}$
$\mathcal{O} \leftarrow \text{OPT}(\mathcal{S})$ ▷ top-level optional fields
$\mathbf{d} \leftarrow \text{CLASSIFY}(\mathcal{M}, p, \mathcal{O})$ ▷ one Y/N call
$D \leftarrow \{f \in \mathcal{O} : d_f = \text{Y}\}$
if mode $=$ DAS **then**
  $K \leftarrow \text{REQ}(\mathcal{S}) \cup D$
  $\sigma \leftarrow \text{SKEL}(\mathcal{S})|_K$
  return $\text{FILL}(\mathcal{M}, p, \sigma)$
else
  $\mathcal{S}' \leftarrow \mathcal{S}$ with REQ $+= D$
  return $\text{DECODE}(\mathcal{M}, p, \mathcal{S}')$
end if

validator-pass rate. *Coverage* is the fraction of oracle-positive pairs the model emits — **equivalent to a field-level recall of demanded fields**, in the slot-filling sense (Henderson et al., 2014; Louvan and Magnini, 2020). *Fabrication* is the fraction of oracle-negative pairs emitted — **a field-level false-positive rate**, the granular analogue of the precision-loss view used by factual-precision measures for free-text generation (Min et al., 2023; Huang et al., 2024). (Structured-output work to date has focused on schema-level *validity* alone; coverage and fabrication operationalise the demand axis that constrained decoding leaves unmeasured.) The oracle is a deterministic regex/synonym matcher, high-precision by construction; the full rule is released. Paired McNemar tests on the grain (sample, field) compare two methods on the same pairs. CIs are paired-bootstrap (Koehn, 2004; Efron, 1979) clustered by *sample* (not pair) (Cameron et al., 2008) so they respect the dependence of multiple options within the sample.

4 The cross-checkpoint pattern

Direction-label thresholds (preregistered). We label each cell (checkpoint, protocol) by $\Delta =$ DAS $-$ SFG in coverage and fabrication: **reversal** (Δcov ≤ -10pp), **mild reversal** ($-10 <$ Δcov ≤ -5pp), **help** (Δcov $\geq +5$pp), **Pareto** ($|\Delta\text{cov}| \leq 5$pp and Δfab ≤ -5pp), **null** (both $|\Delta| < 5$pp), and **otherwise ambiguous** . The thresholds were finalized before the cross-family runs and were notarized in the anonymized release. Every direction label survives its sample-cluster bootstrap 95% CI (Appendix H).

Table 4 reports (Δcov, Δfab) for every cell (checkpoint, protocol); Figure 1 renders the same grid as a colour heatmap of direction labels.

Three patterns are worth noting. *First*, the calibrated frontier checkpoints (Claude Haiku 4.5, Sonnet 4.5; Mistral-7B on the open-weight side) recover 14–24pp of fabrication at near-zero coverage cost on FHIR. *Second*, the per-cell direction varies with protocol in the way the calibration mechanism (§7) predicts: FHIR's Anthropic Pareto corresponds to $|\mu| < 0.05$, while OCPP and OpenAPI sit outside the informative band. *Third*, the Mistral/Llama split on FHIR does not extend beyond the protocol (OCPP: -3.0 vs. -9.1; tools: -6.6 vs. -11.7) and is a property of the demand-classification surface, not the ability.

Paired McNemar at the grain (sample, field) rules out a coverage–fabrication trade-off interpretation: every measured reversal cell has zero correctly emitted DAS-only pairs against 19–31 SFG-only pairs (strict dominance), and Pareto cells show zero DAS-only fabrication pairs against 13–28 SFG-only fabrication pairs ($p < 0.001$).

The benchmark and the calibration regression target the JSON encoding. A cross-serialization audit on six alternative encodings (Appendix G) shows that the JSON-side direction labels are encoding-specific. Two grammar properties of the JSON Schema appear load-bearing. First, the JSON Schema (Wright et al., 2020) represents the optionality of each field as a flag in a dedicated `required` array (Willard and Louf, 2023; Dong et al., 2024); The CGA is then a one-line array edit. YAML, TOML, and S-expression schemas encode optionality through prose conventions, leaving the same intervention without a syntactic handle. Second, JSON's `"key":` delimits each property with two literal tokens that frontier tokenizers segment consistently; the SFG placeholder sits between deterministic delimiters, while YAML's indentation and TOML's = assignments are more ambiguous under a constrained decoder.

A coarse check on the cross-format data is consistent with this picture. The two encodings that admit a machine-checkable `required` predicate yield strongly positive $r(\mu, \Delta\text{cov})$ (JSON $+1.0$, NDJSON $+0.94$), while TOML and S-expressions sit near zero. Python-dict literal at -0.90 is the counterexample: the format has no schema standard at all, and the sign flip suggests a mechanism beyond the hypothesis. A clean isolation of the `required` array property is left to follow-up work.

Checkpoint	FHIR	OCPP	OpenAPI	tools
Open-weight				
Qwen-2.5-3B	$(+1.8, -1.2)$ A	$(-15.6, ?)$ R	$(+24.0, ?)$ H	$(-11.2, -8.2)$ R
Qwen-2.5-7B	$(+6.5, -0.8)$ H	$(-18.8, ?)$ R	$(+53.9, ?)$ H	$(-10.7, +0.0)$ R
Phi-3-mini	$(-12.8, -3.6)$ R	$(-25.0, ?)$ R	$(+29.2, ?)$ H	$(-5.6, -4.9)$ m
Mistral-7B	$(+0.0, -13.8)$ **P**	$(-3.0, +0.0)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(-6.6, +0.5)$ m
Llama-3.1-8B	$(-27.3, -6.0)$ R	$(-9.1, +0.0)$ m	$(-13.3, +0.0)$ R	$(-11.7, +2.2)$ R
OpenAI				
GPT-4o-mini	$(-22.2, -4.3)$ R	$(+0.0, -5.2)$ **P**	$(-25.8, +0.0)$ R	$(-9.7, +0.5)$ m
GPT-4o-full	$(-25.3, -9.5)$ R	$(+0.0, +1.0)$ $\emptyset$	$(+0.8, +0.0)$ $\emptyset$	$(-3.1, -0.5)$ $\emptyset$
Anthropic				
Haiku 4.5	$(-1.0, -24.1)$ **P**	$(-9.1, +0.0)$ m	$(+0.0, +0.0)$ $\emptyset$	$(-6.1, -0.5)$ m
Sonnet 4.5	$(-2.0, -23.3)$ **P**	$(+0.0, +0.0)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(-4.1, +0.0)$ $\emptyset$
Opus 4.5	$(+0.0, -0.9)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(-2.6, +0.0)$ $\emptyset$
Google				
Gemini 3.1 FL	$(+0.0, +6.0)$ A	$(+0.0, +0.0)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(-0.5, +0.0)$ $\emptyset$
Gemma 4 31B	$(-1.0, -0.9)$ $\emptyset$	$(+0.0, +0.0)$ $\emptyset$	$(+0.0, +0.0)^{\dagger}$	$(+0.0, +0.0)^{\dagger}$

Table 4: $(\Delta\text{cov}, \Delta\text{fab})$ in pp for DAS$-$SFG on the 11×4 (checkpoint, protocol) grid. Labels: **P** Pareto, R reversal, m mild reversal, H help, A ambiguous, $\emptyset$ null (§4). "?": open-weight cell without comparable Δfab; †: N=20 subsample.

5 The divergence is not a free-form-prior side-effect

The simplest alternative explanation for the cross-checkpoint divergence in Δ(DAS$-$SFG) is that families differ in how many optional fields they emit zero-shot, so the prune lands on different populations. The data refute this. In FHIR zero-shot (N=57), the seven audited frontier checkpoints emit not just the same raw count but the *same* 82 *specific pairs* of 99: seven-way intersection $= 82$, union $= 84$, pairwise Jaccard $\in [0.976, 1.0]$ (Appendix J). Six of the seven checkpoints match exactly. One detail worth noting: GPT-4o-full, Gemini 3.1 Flash-Lite, and Gemma 4 31B-IT emit *identical* sets on both demanded and undemanded pairs ($82/99$ and $38/116$, symmetric set difference zero in every pairwise comparison) – a three-way exact-set convergence across the OpenAI full tier and both Google checkpoints, despite the absence of a shared training corpus. The 17 commonly missed demanded pairs are all the FHIR-Patient `id` field on prompts that name a patient identifier, which the models treat as redundant with `identifier.value`. The cross-checkpoint divergence in Δ(DAS$-$SFG) therefore lives in the (classifier identity) $\times$ (generator-on-prune compliance) layer, not in the free-form prior.

One Llama-specific caveat deserves a separate check: Llama-3.1-8B emits every optional field on FHIR zero-shot, which could in principle reduce its DAS reversal to an always-Y artifact. The reversal does not shrink under a selective system prompt that drops Llama's free-form rate from 100% to 52.6% — it deepens from $\Delta\text{cov} = -27.3$ to -32.3pp (Appendix J).

6 Random-K mechanism decomposition

To separate mechanism from classifier identity, we compare three modes on the oracle-Y subset across 14 (model, protocol) cells: no promotion (*none*), random selection of k optionals (*R-K*; k matches the classifier's expected positive count), and the demand classifier (*dem.*). We decompose the coverage gain as Δ_{m}=R-K $-$ none (mechanism) and Δ_{c}=dem. $-$ R-K (classifier identity). Table 5 reports a representative slice; the full 14-cell grid is in Appendix I.

Model	Proto	none	R-K	dem.	Δ_{m}	Δ_{c}
GPT-mini	FHIR	38.4	84.5	81.8	$+46.1$	-2.7
GPT-mini	OCPP	51.5	60.6	93.9	$+9.1$	$+33.3$
GPT-mini	tools	1.5	76.0	86.7	$+74.5$	$+10.7$

Table 5: Random-K decomposition (coverage on oracle-Y, %). Δ_{m}= gain from forcing the right *number* of optionals; Δ_{c}= additional gain from picking the right *ones*.

Two regimes emerge. In low-anchor, content-rich schemas (FHIR, OpenAPI) the mechanism component dominates ($\Delta_{\text{m}}=+19$ to $+51$pp; $\Delta_{\text{c}} \leq +4.4$pp). In lexically anchored OCPP and flat function-calling tools, classifier identity contributes a larger share ($\Delta_{\text{c}}= +33.3$pp on GPT-4o-mini OCPP, $+9.7$ to $+10.7$pp on tools). The marginal value of the LLM classifier scales with the within-sample anchor variance, not the mean anchor density of the protocol.

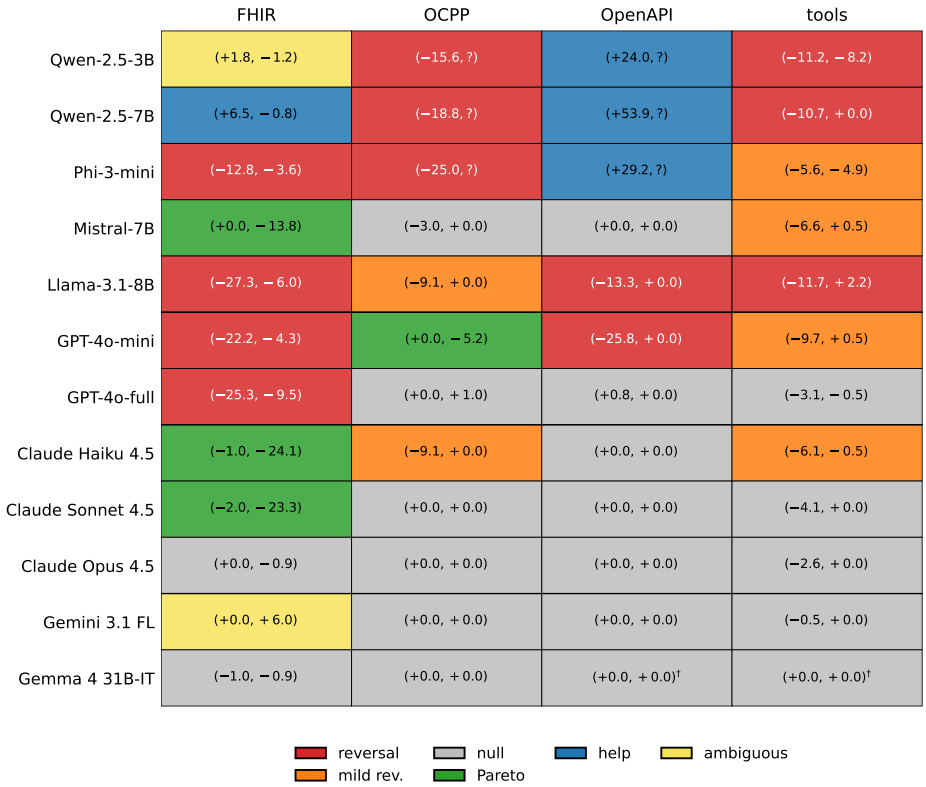

Figure 1: Heatmap of direction labels for DAS−SFG on the 11×4 (checkpoint, protocol) grid; overlaid text is $(\Delta\text{cov}, \Delta\text{fab})$ in pp. †: N=20 subsample.

7 Robustness, downstream, and deployment

Oracle-noise robustness. The coverage and fabrication metrics depend on a deterministic oracle. We flip each label *i.i.d.* with probability $\eta \in \{0, 0.05, 0.10, 0.20, 0.30\}$, average $\Delta(\text{DAS} - \text{SFG})$ over seeds B=20, and report the direction label at every η:

Every direction label is preserved up to η=0.30 on all four cells tested (GPT-4o-mini, Claude Haiku/Sonnet on FHIR; Claude Haiku on OCPP; full table in Appendix D); the qualitative predictions of the framework do not depend on perfect oracle accuracy. As a downstream check, every method achieves 100% HL7 R4 validator acceptance on the spec-clean Patient/Observation subset, despite fabrication differing by 30pp or more between methods — the "fabricated" optional fields are valid FHIR (Appendix N).

Oracle triangulation. Beyond *i.i.d.* noise, we triangulate the surface-form oracle against two LLM judges (Zheng et al., 2023; Chiang et al., 2024; Liang et al., 2023). In FHIR (N=117) the surface oracle disagrees substantially with both the Claude-judge and the GPT-judge ($\kappa = +0.158$ and $+0.236$), while the two judges agree with each other ($\kappa = +0.647$) — consistent with a shared semantic prior over medical-record conventions. The two oracles measure different constructs: *"what did the user ask for?"* versus *"what should the record contain to be well-formed?"*. For demand-aware decoding the surface oracle is the deployment-relevant target; a system that emits every well-formed field reduces to high-fab zero-shot. Under a majority-vote oracle, the GPT-mini and Llama reversals survive (−15.3, −22.6pp); Claude Haiku flips from Pareto to mild-reversal (−10.2pp), the only label change (Appendix K).

Calibration as the cross-protocol predictor. The calibration view we adopt is closest to the binary-label miscalibration studied for LLM confidence (Kadavath et al., 2022; Tian et al., 2023; Guo et al., 2017), applied here at the grain (sample, optional-field). We define classifier miscalibration on a protocol as $\mu \equiv \Pr[Y]_{\text{cls}} - \Pr[Y]_{\text{oracle}}$. $\Pr[Y]_{\text{oracle}}$ **is the mean positive rate of the deterministic oracle rule (§3) applied across the protocol's sample prompts**; it depends on (i) the schema's optional-field set *and* (ii) the sam-

ple prompts, but *not* on any model. Concretely, 0.460 on FHIR, 0.145 on OCPP, 0.779 on OpenAPI, 0.516 on tools.[1] μ is the single observable per cell (classifier, protocol) that we can compute without running DAS, by emitting the classifier on the optional-field set alone.

Two views of μ. We distinguish two $\Pr[Y]_{\text{cls}}$: $\mu_{\text{raw}} \equiv \Pr[Y]_{\text{cls,raw}} - \Pr[Y]_{\text{oracle}}$ uses the classifier's raw Y/N decisions (measures the classifier alone), whereas $\mu_{\text{emit}} \equiv \Pr[Y]_{\text{cls,emit}} - \Pr[Y]_{\text{oracle}}$ uses the rate at which the classifier-marked field actually appears in the DAS output (classifier $\circ$ generator on-prune compliance). The two coincide when the generator strictly follows the prune, and diverge when it silently omits or re-emits marked fields. *The $|\mu| < 0.05$ deployment threshold throughout the paper refers to μ_{emit}*, since the validator-relevant quantity is what gets emitted; we report μ_{raw} as a diagnostic. Example: Claude Haiku FHIR has μ_{emit}=+0.014 (in band) but μ_{raw}=+0.088 (out of band); the 4×4 swap (§8) shows two classifiers with matched μ_{raw} but divergent direction labels, so μ_{raw} alone is insufficient.

We expand to 24 (classifier, protocol) cells, adding Mistral-7B-v0.3, Llama-3.1-8B, and Claude Sonnet 4.5, and we combine $N = 1{,}172$ (sample, classifier, protocol) observations from four closed-API classifiers. Using sample-cluster bootstrapping ($B = 500$, 293 unique clusters), the pooled correlation is $r(\mu_s, \Delta\text{cov}_s) = +0.57$, 95% CI $[+0.46, +0.67]$. Per-protocol: FHIR $+0.49\,[+0.36, +0.59]$, tools $+0.56\,[+0.45, +0.68]$, OCPP $+0.11\,[+0.00, +0.28]$, OpenAPI is degenerate ($\Pr[Y]_{\text{oracle}} = 0.779$ saturates emission). The two informative-regime protocols dominate the signal; the variance of μ is mechanically compressed at the boundaries (Appendix M). Within-protocol 80/20 generalization on FHIR and tools gives $R^2 = 0.14 \pm 0.21$ (CI overlaps zero) and 0.28 ± 0.12 over 10 folds.

The Spearman rank ρ exceeds Pearson in all protocols (FHIR +0.46, OCPP +0.69, tools +0.73, pooled +0.70); OCPP in particular shifts from Pearson +0.11 to Spearman +0.69, indicating a monotone but non-linear relationship (Appendix Q). Plotting $|r|$ against $\Pr[Y]_{\text{oracle}}$ (Appendix O) produces a $\cap$-shape: peak in the informative regime, falloff at the extremes, matching the mechanism's prediction of vanishing effects.

External scope-curve points. We add an external low-density protocol **GraphQL** (GraphQL Foundation, 2021) at $\Pr[Y]_{\text{oracle}} = 0.054$ and a per-R4-type FHIR decomposition with a sub-protocol $\Pr[Y]_{\text{oracle}}$ ranging from 0.20 to 0.60 (Appendix O). GraphQL gives per-sample $r = +0.16$, CI $[+0.01, +0.29]$, consistent with the low-end prediction of the scope condition. Ten points across $\Pr[Y]_{\text{oracle}} \in [0.054, 0.779]$ now populate the scope axis.

Two-stage deployment criterion. (Stage 1) Estimate $\Pr[Y]_{\text{oracle}}$ from the protocol's schema, representative prompts, and the released oracle rule (zero model calls); if it lies in $[0.3, 0.6]$, proceed to Stage 2; otherwise, demand-aware pruning will not change direction, by construction. **(Stage 2)** Run DAS on a small calibration sample (~ 30 samples; one classifier and one generation call per pair), measure $\Pr[Y]_{\text{cls,emit}}$ in the resulting outputs and pick (or distill) a classifier with $|\mu_{\text{emit}}| < 0.05$ as a coarse filter in the expected direction. The protocol-axis check is free; the classifier-axis check costs one small DAS sweep. *Caveat.* $|\mu_{\text{emit}}|<0.05$ is a coarse rate-calibration filter only; we additionally recommend checking Y-precision and Y-recall (≥ 0.6) on the calibration sample to rule out rate-canceling errors (§8).

Lexical baseline ablation (DAS-lex vs DAS-LLM). Does the LLM classifier earn its keep over a zero-cost lexical heuristic? We implement **DAS-lex**: same skeleton-prune pipeline, but the demand classifier emits Y if any token (length ≥ 3) of the field name (camelCase / snake_case split) appears as a whole word in the prompt. The two methods agree on the mean of the protocol μ but disagree on the direction label in 4 **of** 8 FHIR+ tool cells (Appendix P). On FHIR the LLM classifier is strictly better in three of four cells: Claude Haiku DAS-LLM is Pareto (Δcov $= -1.0$pp) while DAS-lex is reversal (-26.7pp); GPT-mini DAS-LLM -22.2pp vs. DAS-lex -82.8pp (catastrophic). On tools the gap narrows because flat function-call field names appear directly in the prompt. DAS-LLM is therefore not a wrapper over lexical matching on content-rich nested schemas; the LLM does semantic work that token overlap cannot replicate.

[1] To make this concrete: for each optional field f in the schema and each prompt p, the oracle assigns a deterministic Y/N from prompt-side surface matching (§3); $\Pr[Y]_{\text{oracle}}$ is the Y-rate averaged over all (sample, field) pairs in the protocol benchmark.

8 Localizing the effect: classifier vs. generator

If the demand-aware direction varies by checkpoint, which component drives it — the classifier's Y/N decisions or the generator's compliance with the pruned skeleton? We separated the two with a 3×3 classifier-by-generator swap in FHIR (N=57) using GPT-4o-mini, Claude Haiku 4.5 and Gemini 3.1 Flash-Lite as the three families (Table 6; full discordant counts and the open-weight extension in Appendix F).

Cls. \ Gen.	GPT	Claude	Gemini
GPT	$(-22.2, -4.3)$ R	$(-13.1, +0.0)$ R	$(-23.2, -4.3)$ R
Claude	$(-3.0, -21.6)$ **P**	$(-1.0, -24.1)$ **P**	$(-3.0, -21.6)$ **P**
Gemini	$(+0.0, +0.0)$ $\emptyset$	$(+0.0, +0.9)$ $\emptyset$	$(+0.0, +6.0)$ A

Table 6: Closed-API 3×3 classifier-by-generator swap on FHIR (N=57). Cell = (Δcov, Δfab) in pp. Classifier row drives direction; discordant counts and open-weight extension in Appendix F (Table F).

The identity of the classifier drives the direction. Swapping the Claude classifier onto the GPT generator turns the canonical FHIR reversal cell into a Pareto cell; swapping the GPT classifier onto the Claude generator destroys Claude's native Pareto. Cross-protocol replication on OCPP and tools (Appendix F) matches the scope condition: outside the informative band the FHIR pattern does not extrapolate.

μ_{raw} is not sufficient (4×4). Adding Llama-3.1-8B as a fourth classifier and generator (Appendix F), Llama-cls and Claude Haiku share *the same* $\mu_{\text{raw}} = +0.088$ in FHIR, yet diverge: Claude-cls rows are uniformly Pareto ($\Delta\text{cov} \in [-4.0, -1.0]$pp), Llama-cls rows uniformly reversal ($\Delta\text{cov} \in [-47.5, -27.3]$pp). The emit-based view separates them ($\mu_{\text{emit}} = +0.014$ for Claude vs. off-range for Llama; Y-precision/recall $0.62/0.90$ vs. $0.39/0.47$). The threshold is on μ_{emit}; the per-sample precision/recall ($P, R > 0.6$) is a recommended sanity check.

9 Discussion

The demand classifier is an instruction-following sub-task that varies across checkpoints at a fixed capability tier; a deployment-time rollout should be validated per-protocol on the target checkpoint. We extend CGA (schema-side variant) to Anthropic `tool_use` and Gemini `response_schema` backends, giving a 30-cell grid across 9 checkpoints (Appendix L). DAS and CGA share a classifier but act through structurally asymmetric levers (Table 7): DAS removes unmarked slots so the generator has no slot to fill; CGA promotes marked slots to `required` so the decoder must emit a value. Under-calibration ($\mu < 0$) causes demanded fields to be dropped under DAS but is benign for CGA; over-calibration ($\mu > 0$) keeps undemanded slots under DAS (the generator may omit) but forces emission under CGA, inflating fabrication.

	DAS (prompt-side)	*CGA (schema-side)*
$f \in D$	keep slot in skeleton	promote to `required`
$f \notin D$	drop slot entirely	leave optional
Under-cal. μ<0	demanded f dropped: **cov loss**	demanded f optional: benign
Over-cal. μ>0	undemanded kept (gen. may omit): mild fab	undemanded forced: **fab inflation**

Table 7: Asymmetric leverage of μ: same classifier output, opposite-signed effect on fabrication.

The CGA's Δfab is positive in 24 of 30 cells. In Claude Haiku FHIR, CGA is Pareto ($\Delta\text{cov} = 0$, $\Delta\text{fab} = -11.2$pp); in GPT-4o-mini FHIR, CGA recovers coverage (-1.0pp vs. DAS's -22.2pp) but pays fabrication of $+43.1$pp. Deployment rule: **(a)** check whether $|\mu_{\text{emit}}| < 0.05$; **(b)** if so and Y-precision ≥ 0.85, use CGA; **(c)** if so but the precision is lower or unknown, use DAS; **(d)** otherwise stay on SFG.

10 Limitations

We note three limitations. (i) *Scale*: FHIR N=57→117 expansion preserves all direction labels (Appendix D). (ii) *Protocol coverage*: the scope curve contains ten points across $\Pr[Y]_{\text{oracle}} \in [0.054, 0.779]$; the ∩-shape remains qualitative at this sample size, and AsyncAPI, gRPC, JSON-LD, and HL7 V2 remain to be added. (iii) *Beyond JSON*: direction labels do not transfer one-to-one to other encodings (Appendix G).

Ethical Considerations

Potential risks. Two risks deserve attention. **(i)** An uncalibrated classifier in safety-critical pipelines (medical FHIR, billing) can *drop* demanded fields (reversal cells in Table 4); pair the $|\mu_{\text{emit}}|$<0.05 filter with Y-precision/recall checks (§7) and human review. **(ii)** Schema validity is not

semantic correctness: 100% HL7 R4 acceptance coexists with 30pp fabrication swings.

References

Marah Abdin and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Anthropic. 2024. Tool use with claude. https://docs.anthropic.com/en/docs/build-with-claude/tool-use.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2023. Prompting is programming: A query language for large language models. In *Proceedings of the ACM on Programming Languages*.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. 2008. Bootstrap-based improvements for inference with clustered errors. *Review of Economics and Statistics*, 90(3):414–427.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *ICML*.

Yixin Dong, Charlie F. Ruan, Yaxing Cai, Ruihang Lai, Ziyi Xu, Yilong Zhao, and Tianqi Chen. 2024. XGrammar: Flexible and efficient structured generation engine for large language models. In *arXiv preprint arXiv:2411.15100*.

Bradley Efron. 1979. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26.

Noam Gat. 2024. LM format enforcer. https://github.com/noamgat/lm-format-enforcer.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *EMNLP*.

Google DeepMind. 2024. Structured outputs in the gemini api. https://ai.google.dev/gemini-api/docs/structured-output.

GraphQL Foundation. 2021. GraphQL specification. https://spec.graphql.org/October2021/.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML*.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franco Pestilli Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? In *Proc. NAACL*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proc. SIGDIAL*.

HL7 International. 2019. Fast healthcare interoperability resources (fhir) release 4. https://www.hl7.org/fhir/R4/.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. In *ACM Transactions on Information Systems*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *EMNLP*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *EMNLP*.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2024. Skeleton-of-thought: Prompting llms for efficient parallel generation. In *Proc. ICLR*.

Brendon Norick and 1 others. 2023. Jsonformer: A bulletproof way to generate structured JSON from language models. https://github.com/1rgs/jsonformer.

Open Charge Alliance. 2017. Open charge point protocol 1.6 (ocpp-j 1.6). https://www.openchargealliance.org.

OpenAI. 2024a. Introducing structured outputs in the api. https://openai.com/index/introducing-structured-outputs-in-the-api/.

OpenAI. 2024b. Json mode & function calling. https://platform.openai.com/docs/guides/structured-outputs.

OpenAPI Initiative. 2024. Openapi specification 3.x. https://www.openapis.org.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. Gorilla: Large language model connected with massive APIs. In *NeurIPS*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *ICLR*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *EMNLP*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proc. ACL*.

Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. In *arXiv preprint arXiv:2307.09702*.

Austin Wright, Henry Andrews, Ben Hutton, and Greg Dennis. 2020. JSON schema: A media type for describing JSON documents. https://json-schema.org/draft/2020-12/json-schema-core.html. Internet-Draft, IETF.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Berkeley function calling leaderboard (BFCL). https://gorilla.cs.berkeley.edu/leaderboard.html.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *EMNLP*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NeurIPS Datasets and Benchmarks Track*.

A Model versions and call dates

All API calls were made between 2026-05-13 and 2026-05-14 KST. Exact version strings:

Alias	API version string
GPT-4o-mini	`gpt-4o-mini-2024-07-18`
GPT-4o-full	`gpt-4o-2024-08-06`
Claude Haiku 4.5	`claude-haiku-4-5-20251001`
Claude Sonnet 4.5	`claude-sonnet-4-5-20250929`
Claude Opus 4.5	`claude-opus-4-5-20251101`
Gemini 3.1 FL	`gemini-3.1-flash-lite-preview`
Gemma 4 31B	`gemma-4-31b-it`

Open-weight checkpoints loaded via HuggingFace Transformers in BF16 with greedy decoding. The complete `environment.json` was recorded per run.

B Demand classifier: prompt and threshold

The demand classifier runs a single LLM call per sample with the system prompt below and a user message that lists the optional fields. The expected output is a concatenated $\{Y, N\}^k$ string where $k = |\text{optional fields}|$ for that sample (capped at $k \leq 12$).

System prompt (verbatim).

```
Decide what fields a user
request requests.  You are
given a USER REQUEST and a
numbered list of OPTIONAL
FIELDS with their types.  For
each field, you answer Y (the
request mentions or contextually
implies a value for it) or N
(the request says nothing about
it).

Example A.
USER REQUEST: Create a Patient
for Marie Curie, female, born
1867-11-07.
OPTIONAL FIELDS: 1.  id
:  string 2.  gender :
enum=[male|female|other|unknown]
3.  birthDate :  string(date)
4.  name :  array 5.  active :
boolean
ANSWER: NYYYN

Example B.
USER REQUEST: Boot a charger
from EVgo model E-100.
OPTIONAL FIELDS: 1.
chargePointSerialNumber :
string 2.  firmwareVersion :
string 3.  iccid :  string
ANSWER: NNN

Output EXACTLY one character per
field, in order, concatenated,
without spaces, no punctuation,
and no prose.
```

Threshold. The mapping from the classifier's Y/N output to the *demanded* set is binary: an optional field is in the demanded set iff its position in the answer string is `Y`. The deterministic decision ($\tau = 0.5$) is what the body reports; a τ-sensitivity ablation is released and replayable at any $\tau \in [0, 1]$.

Per-call budget. One `complete()` call per pair (sample, prompt), with ≤ 64 output tokens. In FHIR this is 57 calls per checkpoint; OCPP 73; OpenAPI 100; tools 100.

Emit-based effective accuracy per classifier. We report the *emit-based effective* agreement $\Pr[\text{DAS-emit} = \text{oracle}]$, the deployment-relevant view that combines the classifier's decision with the generator's on-prune compliance:

Checkpoint	$\Pr[\text{match}]$	$\Pr[Y]_{\text{cls}}$	$\Pr[Y]_{\text{oracle}}$
Mistral-7B-v0.3	0.814	0.637	0.460
GPT-4o-mini	0.665	0.433	0.460
GPT-4o-full	0.679	0.391	0.460
Claude Haiku 4.5	0.819	0.474	0.460
Claude Sonnet 4.5	0.781	0.502	0.460
Claude Opus 4.5	0.749	0.553	0.460
Gemini 3.1 FL	0.712	0.591	0.460

(n=215 pairs each)

Claude Haiku's $\Pr[Y]_{\text{cls,emit}}$ (0.474) is the closest to the oracle's 0.460, consistent with its Pareto label on FHIR.

C Compute budget and API costs

API calls. The complete closed-API audit (GPT-4o-mini, GPT-4o-full, Claude Haiku/Sonnet / Opus 4.5, Gemini 3.1 Flash-Lite) is 11 checkpoints $\times$ {zero-shot, SFG, DAS} in FHIR plus the four-protocol GPT-4o-mini map, plus the 3×3 swap. Per closed-API checkpoint on FHIR: $3\times57 = 171$ generation calls plus 57 demand-classifier calls $= 228$ calls. Total closed-API call count for all runs reported in this paper: $\approx$ 4,600 calls.

Local-GPU compute. Open-weight runs (Qwen-2.5-3B/7B, Phi-3-mini, Llama-3.1-8B, Mistral-7B-v0.3) used a single NVIDIA RTX-A6000 (48GB) in BF16 with greedy decoding via HuggingFace Transformers. The end-to-end wall-clock for the Mistral + Llama sweep cross-family (2 models $\times$ 3 protocols $\times$ 3 generators) was $\approx$ 1.8 hours. Outlines-backed grammar-constrained decoding (CGA on open-weights) is the dominant cost on enum-heavy schemas; the released cache keyed by rewritten-schema JSON content makes repeat sweeps near-free.

D Sample selection and benchmark statistics

Counts. FHIR R4: N=57; OCPP-J 1.6: N=73; OpenAPI 3.x: N=100; tools: N=100. Total N=330 samples. Per-protocol (λ, ν) in Table 2.

Selection criteria. FHIR samples were drawn from the HL7 R4 patient/observation example bundle and filtered to those with (a) at least one required and one optional field, (b) a prompt sentence mentioning at most one optional "trigger" phrase per field, so the oracle can label demand per field deterministically. OCPP and OpenAPI samples are hand-authored prompt-spec pairs covering each schema's top-level operations; tools samples cover the function-calling schemas from the OpenAI tools cookbook plus three synthetic expansions.

Scale: N=57 vs. N=117 expansion. Expanding FHIR from N=57 to N=117 (adding four more R4 types) preserves every direction label: GPT-mini $-22.2 \to -27.7$; Claude Haiku $-1.0/-24.1 \to -3.8/-9.5$; Mistral $+0.0/-13.8 \to +0.0/-10.3$; Llama $-27.3 \to -25.4$.

Oracle label provenance. For each (sample, optional-field) pair, the oracle Y/N label is the surface-form anchor: Y iff the prompt contains a lexical or value-overlap match with the field name. This fixed deterministic oracle is what the η-noise robustness ablation perturbs; see §7.

Oracle-noise robustness (full table). Δ(DAS − SFG) on Δcov (pp) under i.i.d. oracle label-flip noise of rate η, averaged over B=20 seeds:

	η=0	0.05	0.10	0.20	0.30
GPT-4o-mini FHIR	−22	−21	−20	−18	−16
Claude H4.5 FHIR	−1	−2	−4	−6	−9
Claude S4.5 FHIR	−2	−3	−5	−7	−9
Claude H4.5 OCPP	−9	−6	−5	−3	−2

Every direction label is preserved up to η=0.30.

E Generator prompts

This appendix gives the verbatim system and body prompts used by each generator. Each prompt has a one-line system header and a body template that substitutes `{spec:json}`, `{prompt}`, and (for SFG/DAS) `{skeleton:json}`. DAS reuses the SFG fill template; the only difference is that the skeleton has already had optional placeholders the demand classifier marked `N` removed (§3). We include zero-shot, SFG/DAS-fill, and the demand classifier; CGA variants and multi-format renderings are released with the benchmark.

Zero-shot.

```
system: You produce JSON
outputs that strictly conform
to the supplied specification.
Output only JSON, no prose.
body:
Specification: {spec:json}
Request: {prompt}
Output (JSON only):
```

SFG / DAS fill.

```
system: You fill placeholders
in a JSON skeleton with values
from a user request. Output
ONLY JSON. Do not add fields
outside the skeleton. Omit
optional placeholders (marked
"optional") if the request does
not specify them.
body:
Skeleton: {skeleton:json}
Spec (for reference):
{spec:json}
User request: {prompt}
Output a single JSON object.
```

Demand classifier. The classifier system prompt is in Appendix B; the user message lists optional fields and their types, one per line, and asks for a concatenated {Y,N}-string of length k.

F Classifier-by-generator swap: 3×3, 4×4, and cross-protocol

This appendix consolidates all swap-related material: the closed-API 3×3 swap on FHIR with full discordant counts, the cross-protocol replication on OCPP and tools, the 4×4 extension with Llama-3.1-8B, and the per-cell McNemar contingencies. All swaps use DAS; each cell uses the classifier from the row checkpoint and the generator from the column checkpoint.

G Cross-serialization full grid

Figure 2 shows the 5×7×4 cross-format grid (five checkpoints, seven serializations, four protocols) for Δ(DAS − SFG). The JSON-side direction labels do not transfer one-to-one to the other six encodings; two patterns stand out: (a) *the per-cell direction is encoding-specific* (Claude Haiku FHIR-Pareto on JSON becomes mild-reversal

on YAML/XML/S-exp, TOML-Pareto, NDJSON-ambiguous, PyDict-null); (b) *format-direction interaction can be opposite-signed* (Mistral-7B OpenAPI is help on YAML at $(+20, -82)$ but reversal on NDJSON at $(-63, 0)$ and S-exp at $(-58, -18)$); (c) *spread is largest on OpenAPI (high-ν) and smallest on OCPP (high-λ)* (nine of fifteen OCPP cells fall in the null bucket, while OpenAPI cells span all six direction labels).

H Cluster bootstrap 95% CIs for Table 1

Sample-cluster bootstrap 95% CIs are computed for every (checkpoint, protocol) cell with both DAS and SFG measured; B=500–1000 resamples per cell, samples drawn with replacement within each protocol's N $(57, 73, 100, 100$ for FHIR, OCPP, OpenAPI, tools).

Checkpoint	Proto	Δcov [95% CI]	Δfab [95% CI]
Claude H. 4.5	FHIR	-1.0 [$-3.1, +0.0$]	-24.1 [$-35.5, -14.4$]
	OCPP	-9.1 [$-21.9, +0.0$]	$+0.0$ [$+0.0, +0.0$]
	OA	$+0.0$ [$+0.0, +0.0$]	$+0.0$ [$+0.0, +0.0$]
	tools	-6.1 [$-9.5, -3.1$]	-0.5 [$-1.9, +0.0$]
Claude S. 4.5	FHIR	-2.0 [$-5.1, +0.0$]	-23.3 [$-33.6, -14.4$]
	OCPP	$+0.0$ [$+0.0, +0.0$]	$+0.0$ [$+0.0, +0.0$]
	OA	$+0.0$ [$+0.0, +0.0$]	$+0.0$ [$+0.0, +0.0$]
	tools	-4.1 [$-7.0, -1.6$]	$+0.0$ [$+0.0, +0.0$]
Claude O. 4.5	FHIR	$+0.0$ [$+0.0, +0.0$]	-0.9 [$-2.7, +0.0$]
GPT-mini	FHIR	-22.2 [$-31.2, -14.0$]	-4.3 [$-8.0, -0.9$]
	OCPP	$+0.0$ [$+0.0, +0.0$]	-5.2 [$-8.9, -2.7$]
	OA	-25.8 [$-35.0, -18.0$]	$+0.0$ [$+0.0, +0.0$]
	tools	-9.7 [$-13.9, -5.8$]	$+0.5$ [$+0.0, +1.7$]
GPT-full	FHIR	-25.3 [$-32.1, -18.6$]	-9.5 [$-16.0, -2.9$]
	OCPP	$+0.0$ [$+0.0, +0.0$]	$+1.0$ [$+0.0, +2.5$]
	OA	$+0.8$ [$+0.0, +2.5$]	$+0.0$ [$+0.0, +0.0$]
	tools	-3.1 [$-5.8, -1.0$]	-0.5 [$-1.7, +0.0$]
Gemini 3.1 FL	FHIR	$+0.0$ [$+0.0, +0.0$]	$+6.0$ [$+2.4, +10.3$]

OA = OpenAPI.

Every direction label in Table 4 stays on the correct side of its threshold under bootstrap noise.

I Random-K mechanism decomposition: full 14-cell grid

Full 14-cell grid behind §6. Coverage on the oracle-Y subset under three modes: *none* (no promotion), *Rand-K* (random selection of k optionals at the same expected positive rate), and *demanded* (LLM demand classifier). Δ_{mech} = Rand-K − none; Δ_{cls} = demanded − Rand-K.

Model	Proto	none	Rand-K	dem.	Δmech	Δcls
Qwen-2.5-3B	FHIR	62.6	91.6	96.0	+29.0	+4.4
Qwen-2.5-3B	OCPP	93.9	93.9	93.9	0	0
GPT-4o-mini	FHIR	38.4	84.5	81.8	+46.1	−2.7
GPT-4o-mini	OCPP	51.5	60.6	93.9	+9.1	+33.3
GPT-4o-mini	OpenAPI	78.3	82.5	82.5	+4.2	0
GPT-4o-mini	tools	1.5	76.0	86.7	+74.5	+10.7
Mistral-7B	FHIR	73.7	99.0	99.0	+25.3	0
Mistral-7B	OCPP	97.0	100	100	+3.0	0
Mistral-7B	OpenAPI	49.2	100	100	+50.8	0
Mistral-7B	tools	46.4	44.4	54.1	−2.0	+9.7
Llama-3.1-8B	FHIR	66.7	85.9	89.9	+19.2	+4.0
Llama-3.1-8B	OCPP	93.9	97.0	97.0	+3.0	0
Llama-3.1-8B	OpenAPI	64.2	100	100	+35.8	0
Llama-3.1-8B	tools	64.3	69.4	69.9	+5.1	+0.5

Table 8: Random-K decomposition over 14 (model, protocol) cells. Coverage on the oracle-Y subset for three promote-modes: *none* / *Rand-K* / *demanded*. Δmech = Rand-K − none; Δcls = demanded − Rand-K.

J Free-form prior: set-agreement details and Llama selective-prompt control

Full data behind §5. The seven frontier checkpoints (GPT-4o-mini/full, Claude Haiku/Sonnet/Opus 4.5, Gemini 3.1 FL, Gemma 4 31B-IT) audited on FHIR in the zero-shot setting ($N = 57$) emit the same 82 of 99 demanded pairs; pairwise Jaccard $\in [0.976, 1.0]$, seven-way intersection = 82, union = 84. Six of the seven emit identical sets; Claude Opus emits two additional pairs. GPT-4o-full, Gemini 3.1 FL, and Gemma 4 31B-IT match exactly on both demanded (82/99) and undemanded (38/116) pairs with symmetric set difference zero across all three pairwise comparisons.

Llama selective-prompt control. Llama-3.1-8B emits every optional field on FHIR zero-shot, which could in principle reduce its DAS reversal to an always-Y side-effect. Three controls rule this out:

(i) Protocol dependence. Llama's free-form emit rate is 100% on FHIR but only 18.9% on OCPP, 77.9% on OpenAPI, 73.9% on tools — the reversal on the other three protocols cannot follow from always-Y.

(ii) Prompt-induced. A selective system prompt ("Include ONLY the fields explicitly mentioned") drops Llama-FHIR free-form emit from 100% to 52.6%.

(iii) The strongest test. Rerunning both SFG and DAS under the selective prompt gives Δcov = −32.3pp (vs. default −27.3pp) and Δfab = −13.8pp (vs. −6.0pp). The reversal deepens, not shrinks, under the selective prompt; the DAS classifier-side intervention is the proximate cause, not Llama's free-form rate.

K Oracle triangulation: full κ matrix and majority-vote labels

Full data behind §7. Two LLM judges (Zheng et al., 2023; Chiang et al., 2024) score the same (sample, optional-field) pairs as the surface oracle.

κ matrix on FHIR (N=117, 860 pairs).

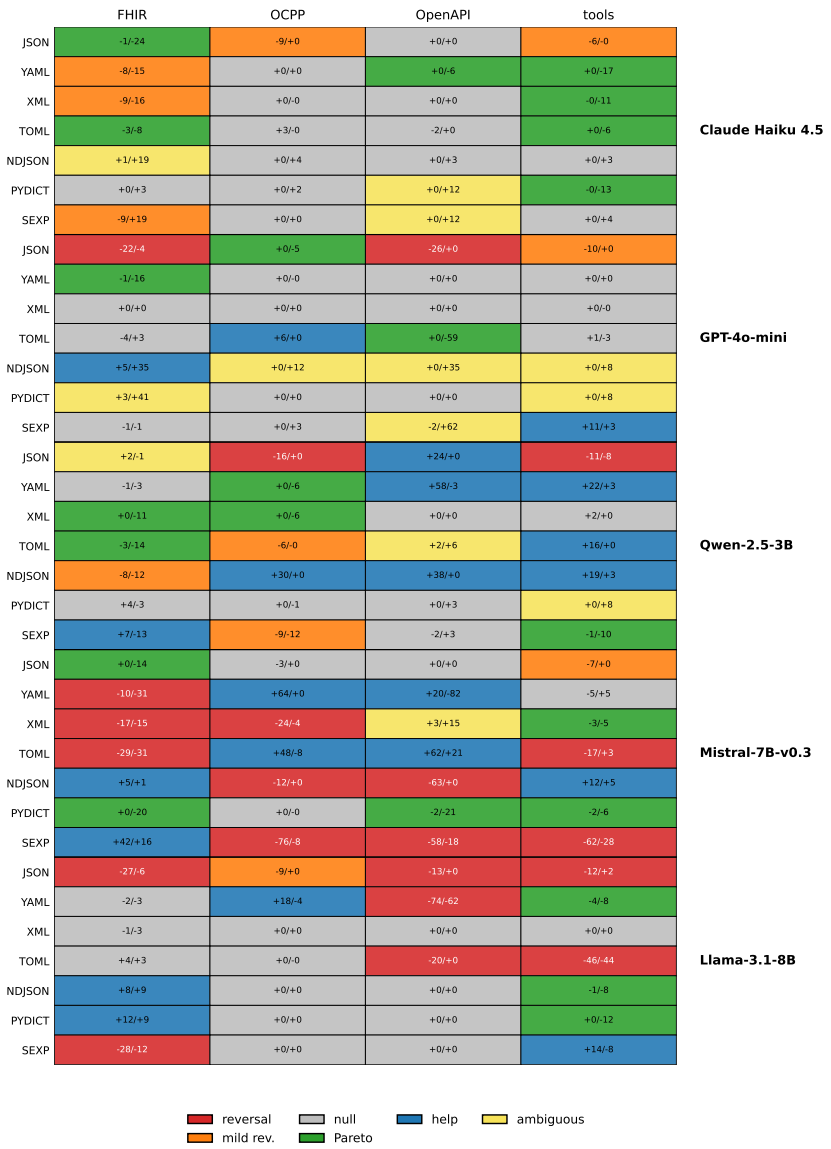

Figure 2: Cross-serialization direction labels for DAS−SFG on the 5-checkpoint × 7-format × 4-protocol grid (N=140 cells). Each row is one (checkpoint, format) combination; each column is one protocol. Cell colour encodes the preregistered direction label; superimposed text is $(\Delta\text{cov}, \Delta\text{fab})$ in pp. Row direction labels differ within a single (checkpoint, format) across protocols (e.g., Mistral-7B YAML: OpenAPI $(+20, -82)$ vs. OCPP $(-24, -4)$).

	surface	Claude	GPT
surface	—	+0.158	+0.236
Claude		—	+0.647

The two LLM judges agree substantially with each other but weakly with the surface oracle. On FHIR the judges call 67–79% of pairs Y vs. surface's 15%; tools sees $\kappa > +0.7$ across all three oracles.

Majority-vote Δcov on FHIR.

Checkpoint	Δcov (maj.)	label
GPT-4o-mini	−15.3	rev. (was rev.)
Mistral-7B	+0.6	null (was null)
Llama-3.1-8B	−22.6	rev. (was rev.)
Claude Haiku 4.5	−10.2	m.rev. (was Pareto)

Three of four labels survive under the most-liberal oracle; Claude Haiku is the one flip. The GPT-mini and Llama reversals are robust across all four oracles (surface, Claude-judge, GPT-judge, majority).

L CGA grid: 30-cell summary and representative slice

Full 30-cell CGA grid behind §9. Below: per-cell Δcov and Δfab for DAS vs. CGA. CGA's Δfab is positive in 24 of 30 cells across the 9 checkpoints; high-precision Anthropic classifiers (Claude Haiku/Sonnet, μ near zero) are the only checkpoints that escape the forced-emission fabrication cost on FHIR.

Checkpoint	Proto	SFG cov	CGA cov	Δcov	Δfab
GPT-4o-mini	FHIR	82.8	81.8	−1.0	+43.1
GPT-4o-mini	tools	98.0	85.7	−12.2	+12.0
GPT-4o-full	FHIR	82.8	68.7	−14.1	+8.6
GPT-4o-full	tools	98.0	93.9	−4.1	+4.9
Claude Haiku 4.5	FHIR	82.8	82.8	+0.0	−11.2
Claude Haiku 4.5	tools	98.0	90.3	−7.7	+6.5
Claude Sonnet 4.5	FHIR	82.8	66.7	−16.2	−27.6
Claude Opus 4.5	FHIR	82.8	76.2	−6.7	−6.9
Llama-3.1-8B	FHIR	79.8	93.9	+14.1	+52.6
Llama-3.1-8B	tools	97.4	69.9	−27.6	+24.5
Mistral-7B-v0.3	FHIR	99.0	100	+1.0	+52.6
Mistral-7B-v0.3	tools	97.4	54.1	−43.4	+19.0

Table 9: CGA versus SFG coverage / Δ on a representative slice of the 9-checkpoint CGA grid (FHIR and tools rows shown for 6 checkpoints; the other 18 cells, mostly null/ambiguous on OCPP and OpenAPI, are in Appendix L). Anthropic and Gemini cells use `tool_use` / `response_schema` as the structured-output enforcement layer.

Checkpoint	Protocol	Δcov	Δfab	Label
GPT-4o-mini	FHIR	−1.0	+43.1	amb.
GPT-4o-mini	tools	−12.2	+12.0	rev.
GPT-4o-full	FHIR	−14.1	+8.6	rev.
GPT-4o-full	tools	−4.1	+4.9	null
Claude Haiku 4.5	FHIR	+0.0	−11.2	**Pareto**
Claude Haiku 4.5	tools	−7.7	+6.5	m.rev
Claude Sonnet 4.5	FHIR	−16.2	−27.6	rev.
Claude Opus 4.5	FHIR	−6.7	−6.9	m.rev
Mistral-7B-v0.3	FHIR	+1.0	+52.6	amb.
Mistral-7B-v0.3	tools	−43.4	+19.0	rev.
Llama-3.1-8B	FHIR	+14.1	+52.6	help
Llama-3.1-8B	tools	−27.6	+24.5	rev.

M Calibration regression: per-protocol table and cell-level estimates

Full data behind §7. Each cell reports the oracle positive rate $\Pr[Y]_{\text{oracle}}$, the classifier's emission rate $\Pr[Y]_{\text{cls,emit}}$ under DAS, their difference μ_{emit}, and the resulting Δcov. Cell-level point estimates (n=6 per protocol) are reported for reference but are under-powered and dominated by the OpenAPI saturated cells; the per-sample cluster-bootstrap values are the primary statistic.

Protocol	point r	cluster-bootstrap 95% CI
FHIR	+0.49	$[+0.36, +0.59]$
OCPP	+0.11	$[+0.00, +0.28]$
OpenAPI	+1.00	degenerate (see below)
tools	+0.56	$[+0.45, +0.68]$

Cls.	Proto	$\Pr[Y]_{\text{or}}$	$\Pr[Y]_{\text{cls}}$	μ	Δcov
GPT-mini	FHIR	.460	.433	−0.028	−22.2
GPT-mini	OCPP	.145	.145	+0.000	+0.0
GPT-mini	OpenAPI	.779	.578	−0.201	−25.8
GPT-mini	tools	.516	.497	−0.018	−9.7
Claude H.	FHIR	.460	.474	+0.014	−1.0
Claude H.	OCPP	.145	.176	+0.031	−9.1
Claude H.	OpenAPI	.779	.779	+0.000	+0.0
Claude H.	tools	.516	.511	−0.005	−6.1
Gemini	FHIR	.460	.591	+0.130	+0.0
Gemini	OCPP	.145	.189	+0.044	+0.0
Gemini	OpenAPI	.779	.779	+0.000	+0.0
Gemini	tools	.516	.542	+0.026	−0.5

Table 10: Per-protocol calibration. $\Pr[Y]_{\text{or}}$ is the oracle positive rate; $\Pr[Y]_{\text{cls}}$ is the classifier's emission rate on optional fields under DAS. μ is their difference.

Protocol	cell-level point r	n
FHIR	+0.82	6
OCPP	−0.03	6
OpenAPI	$+1.00^{\dagger}$	6
tools	+0.37	6
pooled	+0.70	24

† degenerate: $\Pr[Y]_{\text{oracle}} = 0.779$ saturates emission

N HL7 R4 downstream validation

HL7 R4 downstream validation. The fabrication metric counts emitted fields the oracle marks as undemanded, but a fabricated optional can still be valid FHIR. On the spec-clean Patient and Observation subset (n=37 of 57), every method achieves 100% R4 acceptance on every (checkpoint, family) cell, despite `fab` differing by 30pp or more between methods. The fabricated optional fields are valid FHIR.

O Scope curve: figure and extra points (GraphQL, FHIR sub-protocols)

The scope curve plots $|r(\mu_s, \Delta\text{cov}_s)|$ with cluster-bootstrap CIs against $\Pr[Y]_{\text{oracle}}$. The ∩-shape peaks in the informative regime $[0.3, 0.6]$ and falls off at the boundaries, matching the calibration mechanism's prediction.

Filling the curve with a 5th protocol and sub-protocol points. The four pooled protocols give four points on the $\Pr[Y]_{\text{oracle}}$ axis. We add (i) an external protocol **GraphQL** (GraphQL Foundation, 2021) at the extreme low end and (ii) a per-R4-type FHIR decomposition. GraphQL ($\Pr[Y]_{\text{oracle}} = 0.054$, $N = 100$) sits below OCPP at the leftmost edge; the per-sample correlation is $r = +0.16$, 95% CI $[+0.01, +0.29]$ — small in magnitude and consistent with the scope condition's prediction that low oracle density compresses μ's variance. At cell level, all four GraphQL checkpoints

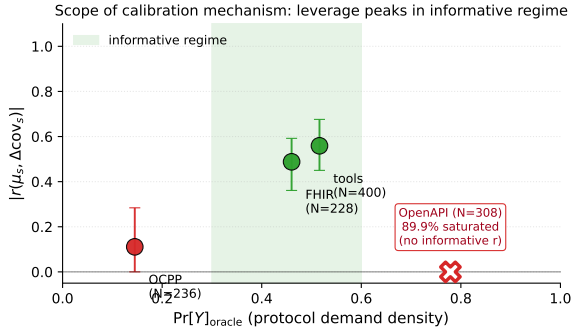

Figure 3: Calibration-mechanism scope curve. The x-axis is the protocol's oracle demand density $\Pr[Y]_{\text{oracle}}$, which can be computed from the protocol's schema and oracle rule with zero model calls. The y-axis is $|r(\mu_s, \Delta\text{cov}_s)|$ with cluster-bootstrap 95% CIs by sample-cluster. The mechanism's leverage peaks in the informative regime $\Pr[Y]_{\text{oracle}} \in [0.3, 0.6]$ (green band): FHIR $r{=}{+}0.49$ CI $[0.36, 0.59]$, tools $r{=}{+}0.56$ CI $[0.45, 0.68]$. At the low extreme OCPP (0.145) the leverage collapses to $r{=}{+}0.11$ CI touching zero. At the high extreme OpenAPI (0.779) the saturation is structural: 89.9% of samples sit at $\mu_s{=}\Delta\text{cov}_s{=}0$, plotted as a degenerate marker. The two-stage deployment criterion follows directly from this curve.

show reversal labels (GPT-mini -33.3, Claude Haiku -11.1, Mistral -14.8, Llama -18.5pp): DAS uniformly hurts coverage when the classifier prunes $\approx$95% of optionals. The five FHIR R4 sub-protocols yield Observation ($\Pr[Y]_{\text{oracle}} = 0.362$), MedicationRequest (0.300), Encounter (0.500), Patient (0.600), DiagnosticReport (0.200); per-sub-protocol r values are small (-0.06 to $+0.27$) because within-type μ has narrow variance, but their $\Pr[Y]_{\text{oracle}}$ values place each on the scope axis where the pooled curve predicts. The scope $\cap$-shape lives at the protocol-level granularity at which the deployment criterion operates, with 10 points across $\Pr[Y]_{\text{oracle}} \in [0.054, 0.779]$ now populating the axis.

P DAS-lex vs DAS-LLM: per-cell ablation

We compare the LLM-based DAS classifier against two zero-cost lexical heuristics on the FHIR and tool cells. **DAS-lex-tok** splits each field name into camelCase/snake_case tokens and emits Y if any token (length ≥ 3) appears as a whole word in the prompt. **DAS-lex-sub** uses raw-substring matching.

Ckpt	Proto	Δcov (LLM)	Δcov (lex-tok)	Δcov (lex-sub)
GPT-mini	FHIR	-22.2	-82.8	-82.8
GPT-mini	tools	-9.7	-5.1	-24.0
Claude H.	FHIR	-1.0	-26.7	-17.4
Claude H.	tools	-6.1	-5.1	-15.8
Mistral	FHIR	$+0.0$	-9.0	—
Mistral	tools	-6.6	-0.5	-2.6
Llama	FHIR	-27.3	-44.4	-35.2
Llama	tools	-11.7	-4.6	-14.3

DAS-LLM and DAS-lex differ in the direction label in 4 of 8 cells. In FHIR the LLM is strictly better in three of four cells; the gap is largest where the prompt names a property semantically (e.g. "Marie Curie, born 1867") without naming the field token (`birthDate`). In tools where the function-call schema field names appear directly in the prompt, the lexical heuristic narrows.

Q Spearman robustness on the calibration regression

Spearman Per-protocol $\rho(\mu_s, \Delta\text{cov}_s)$ with cluster-bootstrap CI ($B{=}500$).

Protocol	n	Pearson r	Spearman ρ
FHIR	228	$+0.49$ $[+.36, +.60]$	$+0.46$ $[+.29, +.62]$
OCPP	236	$+0.12$ $[+.04, +.30]$	$+0.69$ $[+.47, +.85]$
OpenAPI	308	$+1.00$ degen.	$+1.00$ degen.
tools	400	$+0.56$ $[+.45, +.66]$	$+0.73$ $[+.57, +.86]$
joint (F+t)	628	$+0.50$	$+0.59$
pooled	—	—	$+0.70$

Spearman exceeds Pearson on every protocol; OCPP in particular shifts from a near-zero Pearson ($+0.11$) to a strong monotone Spearman ($+0.69$), implying a non-linear but monotone calibration–coverage relationship. lationship. lexical heuristic narrows the gap to within 1–3pp.

R Artifact licenses and intended use

Models. We use models under the following licenses and intended use statements as documented by their creators (accessed 2026-05). Open weights: Qwen-2.5-3B/7B (Apache 2.0); Phi-3-mini-4k (MIT); Mistral-7B-Instruct-v0.3 (Apache 2.0); Meta-Llama-3.1-8B-Instruct (Llama 3.1 Community License). Closed APIs: GPT-4o-mini, GPT-4o (OpenAI Services Agreement); Claude Haiku/Sonnet/Opus 4.5 (Anthropic Usage Policies); Gemini 3.1 Flash-Lite (Google Generative AI Additional Terms). Each license permits research use and publication of evaluation results, which is how the models are used here.

Schemas and standards. FHIR R4 ((HL7 International, 2019)) is released by HL7 under the CC0 1.0 license. OCPP-J 1.6 ((Open Charge Alliance, 2017)) is published by Open Charge Alliance under

their open specification policy. OpenAPI 3.x ((OpenAPI Initiative, 2024)) is released under Apache 2.0 by the Linux Foundation. GraphQL ((GraphQL Foundation, 2021)) is published under the OWFa 1.0 license. The synthetic Petstore OpenAPI document we use is derived from the publicly released Petstore example (Apache 2.0). All uses here are research-only evaluation, consistent with the intended uses stated by each standard body.

Released benchmark and code. The 330-sample (FHIR, OCPP, OpenAPI, function-calling) benchmark, the deterministic oracle rule, the synonym list, the demand-classifier prompt, all skeleton renderers, and the evaluation pipeline are released under the MIT license at the anonymized URL in the submission. Intended use is research on demand-aware structured decoding; redistribution is permitted under MIT.

S Data: PII, offensive content, and human subjects

PII and identifiability. FHIR samples are drawn exclusively from the publicly released HL7 R4 example bundle and from synthetic Petstore-style resources; no real patient records are included. Names, identifiers, dates of birth, and addresses are the synthetic example values shipped by HL7 (e.g., `Patient/example`, `Patient/pat1`), which are designed for testing and contain no real individuals. OCPP samples cover synthetic charging-station transactions with no PII. OpenAPI samples cover a synthetic Petstore product catalog. Function-calling samples cover synthetic web-search, mail, calendar, and database operations; the body text never names a real individual.

Offensive content. The benchmark prompts and responses are domain-restricted to structured back-office tasks (medical record creation, charge-point telemetry, REST API parameter filling, agent tool calls). We manually reviewed all 330 prompts and the free-form zero-shot completions from 11 checkpoints, and found no offensive, hateful, or unsafe content.

Human subjects. The work uses no human annotators, no crowdworkers, and no participant recruitment. Every label in the benchmark is produced by the deterministic surface-form oracle described in §3; the LLM-judge triangulation (§7) uses model-generated, not human, judgments.

T Hyperparameters and packages

Generation hyperparameters. All API calls and all open-weight runs use temperature 0 (greedy decoding), top-$p = 1$, top-$k = 0$ (disabled), and the default per-API maximum-token limit. The demand classifier is capped at ≤ 64 output tokens. Open-weight models are loaded in BF16 via HuggingFace Transformers 4.43. We do *not* perform any hyperparameter search: the calibration regression and the $|\mu_{\text{emit}}| < 0.05$ threshold are reported on the first (and only) configuration we ran.

Constrained-decoding backends. Open-weight CGA uses Outlines 0.1.4 with the JSON-schema backend. OpenAI CGA uses the `response_format={type:"json_schema"}` structured-outputs API. Anthropic CGA uses the `tool_use` API with the schema as a single tool. Gemini CGA uses `response_schema` in the `generationConfig` field. All four backends were used with their default stop conditions.

Statistical packages. Paired McNemar tests use `statsmodels` 0.14. Cluster bootstrap uses our own NumPy implementation with B=500 resamples clustered by sample. Spearman and Pearson correlations use `scipy.stats` 1.13.