EVIDENCE-TO-DELIVERY PROTOCOLS FOR CONSISTENT MULTI-ARTIFACT DATA ENGINEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM pipelines for data engineering can generate individually plausible artifacts—models, schemas, mappings, and execution plans—without ensuring they form a coherent delivery package. This gap motivates an evidence-to-delivery framework that separates and tests two mechanisms: candidate generation and candidate selection. We propose PlanGate, which shares one relation plan across eight requested artifacts and delegates DDL admission to a reference-free deterministic predicate. To isolate how relational evidence affects each protocol, we introduce a factorial intervention over foreign-key declarations and meaningful column names, combined with matched planned/unplanned streams for generation effects and fixed-pool replays for selection effects. We also present DE-VeriBench, 306 cases across seven source packages, evaluating structural fidelity, parsing, execution, and package consistency as distinct outcomes. Across seven models from four model families, PlanGate achieves higher mean foreign-key topology recovery than Single-shot for every model. Crucially, the joint analysis distinguishes a better structural proposal from executable DDL and from a fully verified package. This framework gives data engineers an explicit basis for deciding what information to preserve, where to allocate inference, and which properties must be checked before submission.

1 INTRODUCTION

In the realm of data engineering, the interplay between modeling, integration, transformation, loading, and validation is paramount, with each stage dependent on the deliverables of the others. This intricate dance is further complicated by the need for these stages to be coordinated through declarative interfaces, as seen in systems like LOTUS and Palimpzest (Patel et al., 2025; Liu et al., 2025a). The challenge lies in ensuring that the relationships encoded in one artifact, such as a logical data model, are accurately reflected across all related artifacts, including schema DDL, entity–relationship diagrams, data dictionaries, source mappings, transformation rules, load dependency graphs, and validation tests.

Recent advancements in schema design (Wang et al., 2025; Sadia et al., 2025) have not fully addressed this coherence challenge. For instance, a relationship like `orders.customer_id` referencing `customers.id` must be consistently represented across all these artifacts. The mere plausibility of individual outputs does not guarantee coherence across the entire set of deliverables. This coherence gap is the focal point of our research.

To bridge this gap, we introduce an evidence-to-delivery framework that disentangles the processes of candidate generation and selection. This framework is underpinned by mechanisms such as planning, self-criticism, debate, verification, and retries, which can influence both the generation and selection of candidates (Wang et al., 2023; Shinn et al., 2023; Madaan et al., 2023; Du et al., 2024; Ni et al., 2023). By separating these roles, we can more effectively allocate computational resources and refine the SQL selection process (Snell et al., 2025; Pourreza et al., 2025).

Our framework operates in three interconnected layers: the available relational evidence, the coordinated proposal of deliverables, and their independently evaluated acceptance. At the heart of this framework is PlanGate, which utilizes a fixed relation plan to guide the generation or regeneration of all eight deliverables. A reference-free predicate determines the admission of DDL, ensuring that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

only valid and sufficiently referenced tables are accepted. This setup allows for a clear separation of the coordination mechanism and the submission boundary, making both aspects inspectable.

To assess the impact of relational evidence on the generation and selection processes, we employ a pre-registered factorial intervention that manipulates foreign-key declarations and column names. This intervention helps us understand whether one form of representation can compensate for the lack of the other. By using matched planned/unplanned streams and fixed-pool replays, we can estimate the effects of generation and selection, respectively.

For a comprehensive evaluation, we turn to DE-VeriBench, a benchmark comprising 306 cases from seven source packages. This benchmark enables us to measure structural quality, naming effects, execution, package consistency, and resource use as distinct engineering outcomes. Sections 4 and 5 of our work delve into the definition and evaluation of these measurements.

Our contributions are threefold:

- **The Evidence-to-Delivery Framework:** PlanGate exemplifies how a shared relational context and deterministic DDL admission can be combined to ensure coherent delivery across artifacts. Through interventions and fixed-candidate controls, we can discern the roles of evidence use, generation, and selection.
- **DE-VeriBench Evaluation:** This benchmark provides a common task contract, allowing us to measure structural fidelity, parsing, execution, package consistency, and other engineering outcomes in a consistent manner.
- **Protocol-Dependent Outcomes:** Our comparative analysis of seven models from four families reveals that PlanGate consistently outperforms Single-shot in foreign-key topology recovery.

By presenting this framework and its evaluation, we offer data engineers a structured approach to deciding what information to preserve, where to allocate inference, and which properties must be verified before submission, thereby enhancing the reliability and coherence of data engineering pipelines.

2 Related Work

Data-engineering automation has progressively moved from isolated schema or query generation toward representations that connect sources, targets, and dependencies. Data exchange formalizes these correspondences (Fagin et al., 2005); Magneto uses retrieval and LLM reranking for schema matching (Liu et al., 2025b); Text2Schema/SchemaAgent and SQUiD generate schemas or populated tables from textual descriptions (Wang et al., 2025; Sadia et al., 2025); and LOTUS and Palimpzest expose declarative, quality-aware LLM data processing (Patel et al., 2025; Liu et al., 2025a). Database benchmarks such as Spider, BIRD, Spider 2.0, and EvoSchema evaluate query or schema-evolution behavior (Yu et al., 2018; Li et al., 2023; Lei et al., 2025; Zhang et al., 2025b). DE-VeriBench complements these settings by applying one eight-artifact contract and a frozen evaluator to 306 cases from heterogeneous source packages, and by reporting structural recovery, parsing, execution, and package consistency as separate outcomes rather than collapsing them into a single score. Its multidimensional reporting follows established data-quality and benchmark-validity principles (Pipino et al., 2002; Bowman & Dahl, 2021) and the broader practice of reporting multiple capabilities and resource budgets (Liang et al., 2023; Chan et al., 2025).

Inference-time planning and selection provide the second relevant line of work. Least-to-most prompting, self-refinement, debate, multi-agent workflow systems, and workflow search organize generation through decomposition, critique, collaboration, or workflow search (Zhou et al., 2023; Madaan et al., 2023; Du et al., 2024; Wu et al., 2024; Hong et al., 2024; Zhang et al., 2025a). Adapting inference computation to problem difficulty provides a complementary control (Snell et al., 2025); CHASE-SQL illustrates candidate diversity followed by learned selection (Pourreza et al., 2025). DB-GPT and Data-Copilot are adjacent database-oriented workflow systems (Xue et al., 2023; Zhang et al., 2023); here their abstractions are adapted to the same eight-artifact contract and frozen evaluator, without claiming native system equivalence. Sketching and solver-aided languages separate a proposed structure from executable constraints (Solar-Lezama et al., 2006; Torlak & Bodík, 2014), while PICARD and Outlines constrain decoding (Scholak et al., 2021; Willard

& Louf, 2023), Synchromesh enforces constraints through completion engines (Poesia et al., 2022), and LEVER and Self-Debugging use execution-aware verification or feedback (Ni et al., 2023; Chen et al., 2024). PlanGate instead admits a complete package through a deterministic post-generation DDL predicate and regenerates under the same relation plan; matched planned/unplanned streams and frozen-pool replays consequently make generation and selection effects separately inspectable in a cross-artifact task.

3 Architecture and Experimental Design

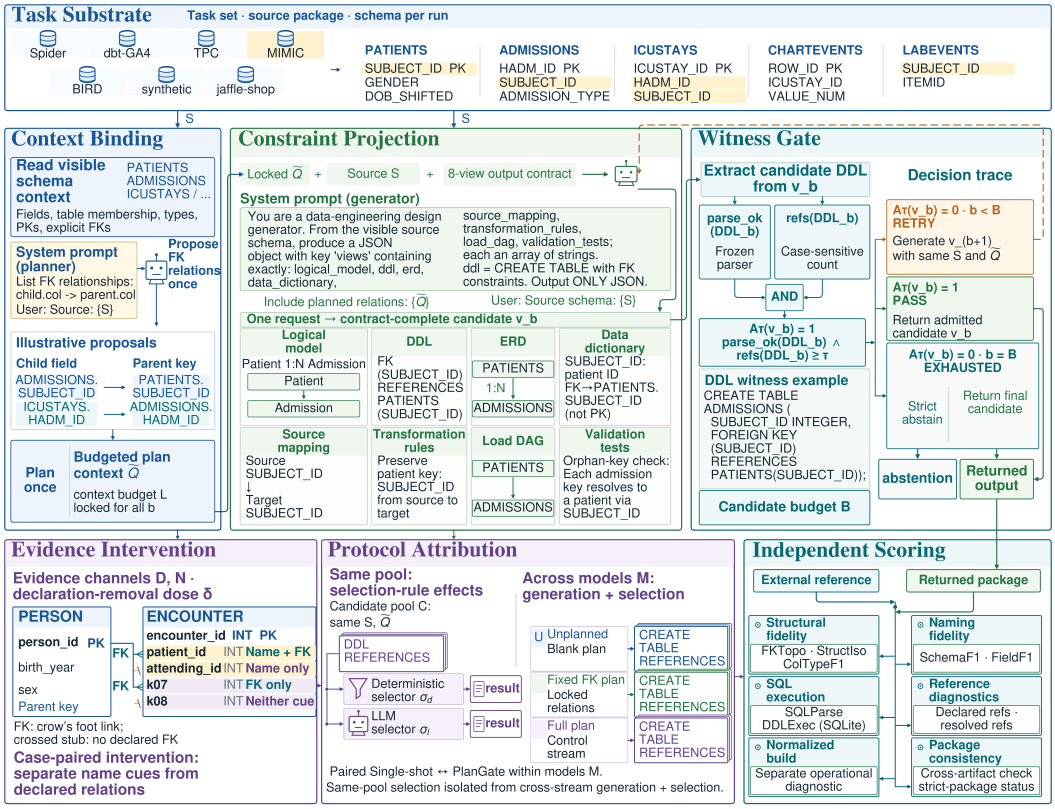

Figure 1: PlanGate architecture. A relation plan is computed once from the visible schema and shared across eight-artifact generation attempts. The DDL gate checks parseability and explicit relation witnesses, then accepts the candidate or requests another attempt under the same plan. Exhaustion follows the recorded fallback rule for each execution track (Appendix A), not gate admission. The illustrated FK is a proposed relation.

The architecture in Figure 1 links five stages into one auditable protocol: the visible source schema supplies relational evidence; the planner compresses that evidence into one shared relation plan; a joint generator realizes the eight requested views; the deterministic witness gate sets the DDL submission boundary and, when needed, requests another realization under the same plan; and the frozen evaluator scores structure, execution, and cross-view consistency after generation. Evidence interventions and attribution controls are analysis layers around this path, not additional generation modules. Thus the planner and generator cooperate through the locked plan, whereas admission and scoring remain separate checks with different access to the reference package.

3.1 Task Substrate

The task maps visible source-schema text S to a package specified by the fixed eight-artifact contract Ω_8. The ordered package contains a logical model, schema DDL, ERD, data dictionary, source mapping, transformation rules, load DAG, and validation tests. These coupled outputs constitute one

package: its DDL defines the database, while the other artifacts describe its entities, dependencies, source correspondences, and operational checks. The generator returns them as named fields of one JSON object. Both compared protocols receive the same source schema and artifact request. The reference package is reserved for post-hoc scoring; generation and admission use only the visible input and the candidate itself.

The output contract makes the coordination problem concrete. A top-level `views` object contains eight named arrays of strings; DDL entries request `CREATE TABLE` statements with FK constraints. Diagrams, mappings, load dependencies, and tests remain textual views rather than separate executable languages enforced during decoding. The 65 dbt cases supply model SQL/source text as S, while the other 241 cases supply source DDL. This common interface permits joint generation across both input forms.

3.2 Context Binding

The planner reads the complete visible schema, including tables, column names, types, primary keys, and any declared foreign keys. It proposes a free-text relation plan Q using the instruction `List FK relationships: child.col -> parent.col`. We distinguish this proposal from the text actually passed to the generator:

$$\widetilde{Q} = T_L(Q) = Q[1 : \min(L, |Q|)], \tag{1}$$

where $|Q|$ is the number of characters and L is the plan-context budget. The primary protocol reuses the same fixed prefix $\widetilde{Q}$ across generation attempts. The source schema itself remains available in full, so the prefix limits only the additional planning signal. The registered prefix length, transport budgets, and length-sensitivity settings are specified in Appendix A. The illustrative relation `ADMISSIONS.SUBJECT_ID -> PATIENTS.SUBJECT_ID` in Figure 1 comes from a MIMIC-derived seven-table schema without declared FKs or clinical records. Sharing the plan exposes the same proposed dependencies to each generation attempt. A fixed L does not preserve an equal fraction of relations across schemas; plan-length and budget sensitivity are evaluated separately.

3.3 Constraint Projection

The generator receives S, the locked $\widetilde{Q}$, and Ω_8, and returns one JSON package per attempt. Retries regenerate the package under the same source, without a new plan or repair prompt. Each planned relation is intended to align DDL constraints with ERD edges, dictionary keys, mappings, transformations, load order, and tests. The gate checks only the DDL witness; independent evaluation measures the broader artifact relationships.

For example, the plan `orders.customer_id -> customers.id` asks for an FK declaration in DDL, the corresponding ERD edge, a dictionary key annotation, and a source-to-target key mapping. Transformations should preserve the mapped key, the load DAG should place the parent before the child, and validation tests should check the dependency. This illustrative correspondence shows what shared conditioning contributes: one relational proposal guides several representations. It is an instruction to the joint generator, not eight independently enforced constraints. Appendix A gives the exact prompt and a worked acceptance example.

3.4 Witness Gate

The deterministic gate extracts candidate DDL and decides admission separately from lenient fallback return. For a candidate v and threshold $\tau \geq 1$,

$$A_\tau(v) = 1 \iff \mathrm{parse_ok}(v.\mathrm{DDL}) = 1 \ \wedge \ \mathrm{refs}(v.\mathrm{DDL}) \geq \tau. \tag{2}$$

Here, $\mathrm{parse_ok}$ indicates that the frozen SQL parser recovers at least one table, and refs counts occurrences of the case-sensitive string `REFERENCES` in the registered live protocol's DDL. The gate reads no reference graph and invokes no LLM. The plan conditions generation but is not checked edge by edge by this predicate.

The minimal protocol uses an FK-only plan, $\tau = 1$, and at most $B = 3$ candidate generations. Let $b^\star$ be the first passing index, or $B + 1$ if no candidate passes. The primary seven-model comparison

returns

$$v_{\text{return}} = \begin{cases} v_{b^\star}, & b^\star \le B, \\ v_B, & b^\star = B + 1. \end{cases} \tag{3}$$

Both service interfaces use this final-slot fallback. Every case remains in the scoring denominator, with empty returns handled by the registered scoring convention; strict semantics instead abstain on exhaustion. Thus an admitted output and a returned output are distinct events. For the example above, an ERD edge is evaluated through the DDL witness: a parseable DDL containing the registered `REFERENCES` token meets this predicate, while agreement with the other seven views still requires independent evaluation. Relation-free DDL and lowercase-only declarations fall outside this lexical predicate and remain governed by the registered return rule. Full prompts, serialization rules, and the fixed-budget variants are specified together in Appendix A.

3.5 Evidence-Channel Intervention

Let $D \in \{0, 1\}$ indicate whether explicit foreign-key declarations are visible and $N \in \{0, 1\}$ indicate whether semantically meaningful column names are visible. We test all four (D, N) cells by deleting declarations and/or replacing column identifiers with opaque tokens. The primary structural outcome is FKTopo, a signature-conditioned FK-edge F1 sensitive to table and column context. It compares reference and generated edges after table signatures are anonymized; copied names alone cannot earn credit.

A single-factor ablation cannot distinguish an uninformative declaration from a declaration whose removal is compensated by meaningful names. The four cells identify each channel's conditional effect and their interaction. For example, the declaration effect with meaningful names is $Y_{11} - Y_{01}$, and the interaction is $(Y_{11} - Y_{01}) - (Y_{10} - Y_{00})$, where Y_{dn} is the mean outcome in cell (d, n). Table names, types, and model priors remain available in all four conditions; the comparison isolates the two manipulated input channels under this shared context.

Masking precedes planning, so the planner and generator receive the same transformed source. The inverse column-renaming map is used only on generated DDL for scoring, never as generation context, and is derived from source identifiers rather than reference relations. This prevents the planning stage from restoring a deliberately hidden name through an unmasked input.

Effects are paired by case within the supported transformations: 168 cases support declaration removal and the full factorial comparison, while 246 support joint masking. A residual-context control additionally anonymizes table and constraint names and removes comments, retaining types and schema structure. It tests what useful context remains beyond the two manipulated channels. Complete estimates and uncertainty are in Appendix D.

A five-level declaration-removal experiment resolves effects beyond the binary endpoints. For m parsed declarations and nominal percentage $p \in \{0, 25, 50, 75, 100\}$, it removes $k = \lfloor (pm + 50)/100 \rfloor$ declarations, rounding half-integers upward. The first k declarations in a case- and seed-specific shuffled order are removed, so the doses are nested. Slopes use the actual parsed removal fraction; identical transformed inputs are deduplicated before estimation. This matters for small schemas, where distinct nominal percentages can produce the same input rather than independent evidence.

3.6 Protocol Attribution

The main comparison pairs Single-shot and PlanGate for seven models on 306 cases, yielding 2,142 within-model pairs and 4,284 final outputs. Cases are the statistical unit. Service-specific output budgets, JSON retries, and selection settings make this a deployed-protocol comparison: PlanGate can use one planning call and up to three candidate generations, whereas Single-shot requests one candidate.

Separate controls use unplanned (U) and planned (P) candidate streams. For each case, let $Y_{s,1}$ be first-candidate FKTopo in stream s and $Y_{s,g}$ the outcome selected by the same gate from that stream. We estimate

$$\delta_{\text{plan}} = Y_{P,1} - Y_{U,1}, \qquad \delta_{\text{gate},s} = Y_{s,g} - Y_{s,1}, \tag{4}$$

$$\delta_{\text{interaction}} = \delta_{\text{gate},P} - \delta_{\text{gate},U}. \tag{5}$$

Averaging these case-level differences distinguishes a change in generated candidates from a change due to selection. The LLM-judge control instead holds each three-candidate pool fixed and changes only the selector; the judge adds a nominal model call, so identical candidates do not imply identical compute. Other controls vary FK-only versus full plans, prefix length, plan reuse, and shared versus separate artifact plans. Appendices E and F provide configurations and complete contrasts.

4 DE-VERIBENCH

Cases and intended use. DE-VeriBench is the benchmark introduced in this paper for evaluating multi-artifact database design. Unlike query-only or schema-only benchmarks, it holds a common eight-artifact contract and a frozen evaluator across heterogeneous source packages, so structural recovery, parsing, execution, and cross-artifact package checks can be reported as separate outcomes. Its 306 cases comprise Spider (166), dbt-GA4 (60), TPC-derived specifications (36), MIMIC schema/governance tasks (18), BIRD (11), synthetic stress cases (10), and dbt jaffle-shop (5). Every case instantiates the same Ω_8 output contract, yielding 2,448 required artifact instances. Reference designs contain 1,285 tables, 7,010 columns, and 3,661 recorded validation assertions across the 306 cases. These are schema and test specifications, rather than stored-data volumes or observed test executions.

The source input describes the existing system; the reference specifies the target design used for scoring. Complete source DDL in 241 cases contains 1,220 tables and 6,608 columns. The remaining 65 dbt cases supply model SQL rather than complete source DDL. Appendix B breaks down these totals by source and recorded domain label.

The primary comparison evaluates both protocols on the same 306 cases with equal case weighting. The development, test, and stress labels contain 183, 92, and 31 cases; no protocol fitting uses the development subset. Exact source/reference deduplication yields 296 clusters, with three duplicate pairs crossing these labels. Case adaptation used automated construction and agent checks; reference artifacts retain their source-benchmark provenance. Appendix B documents provenance, duplication, and source-specific release terms.

Structural and naming fidelity. Evaluation covers structural fidelity, naming agreement, SQL executability, and consistency across the eight artifacts. The three primary structural metrics are task-specific multiset F1 measures that remove direct identifier reward. StructIso compares multisets of table signatures (column count, canonical types, primary-key presence, and FK out-degree), not exact graph isomorphism. FKTopo compares edges between these signatures when both endpoints resolve; ColTypeF1 compares column-type multisets. Empty returns follow the registered scoring convention. SchemaF1 (also called RawSchemaF1) matches table names; FieldF1 matches data-dictionary field names. These are secondary naming-sensitive diagnostics. FKTopo gives one when both resolved edge multisets are empty and omits edges whose endpoints lack local signatures. We therefore report declared-reference and resolved-edge diagnostics alongside it, including 224 declaration-bearing references and 222 references with nonempty resolved graphs.

Calibration. Two zero-model-call checks support the current claims: removing all FK declarations lowers FKTopo from 1.000 to 0.009 on 224 applicable references, with the residual coming from two empty resolved graphs, whereas complete deletion gives zero on the 222 references with nonempty resolved graphs. Arbitrary renaming leaves StructIso unchanged while lowering the name-sensitive diagnostic. These reference-based checks are independent of model generation; the remaining reference-side perturbations and scorer diagnostics are listed in Appendix C.

Execution and package consistency. SQLParse measures SQLGlot parsing success with at least one extracted table. DDLExec (SQLite) additionally requires the emitted script to execute in an in-memory SQLite database. The seven-model package evaluation separately reports schema-build success after SQLGlot normalization to SQLite. Raw-script execution and normalized builds use different dialect handling and are reported separately; parse-only checks are identified as parsing proxies. The architecture's SQL execution box groups these two endpoints: SQLParse is the parse proxy, whereas DDLExec tests execution of the original script. Within each evaluation, both protocols share the evaluator and case set.

Package evaluation checks dictionary/schema agreement, column-level relationships, mappings, load dependencies, and executable tests across the eight artifacts. Strict package success requires all artifacts and all applicable checks to pass; unsupported representations are reported through parsing coverage (Appendix C.4). This distinguishes structural resemblance from executable, internally consistent deliverables. PlanGate itself observes only parsing and declared-reference count, not these independent evaluation outcomes.

5 EXPERIMENTS

We compare seven registered models and four adapted protocol conditions using the same DE-VeriBench 306-case corpus and the same frozen evaluator. The four conditions are the two external workflow adaptations, DB-GPT and Data-Copilot, each run with GPT-5.6 Sol and GPT-5.5; they are reported alongside the two registered protocols, Single-shot and PlanGate, for endpoint calibration rather than as native-system replicas. Single-shot requests one package; PlanGate adds planning and bounded regeneration. The main comparison therefore measures deployed protocol behavior under the registered budgets, while separate interventions and fixed-candidate controls identify evidence, generation, and selection effects (Appendices D and F).

5.1 FULL-CORPUS QUALITY AND ENDPOINT SEPARATION

The five endpoints in Table 1 separate delivery stages: **FKTopo** is naming-independent recovery of resolved FK edges; **StructIso** is F1 over table-signature multisets (column count, canonical types, primary-key presence, and FK out-degree), not exact graph isomorphism; **ColTypeF1** is F1 over canonical column-type multisets; **SQLParse** is the fraction of outputs in which the DDL parser recovers at least one table; and **DDLExec** is the fraction of original scripts that execute in in-memory SQLite with FK checks enabled. The first three are structural endpoints, followed by parseability and direct execution. Operationally, FKTopo measures dependency structure at the table-signature level, providing a proxy relevant to joins, referential checks, and parent–child load ordering. StructIso measures the distribution of entity shapes through column counts, canonical types, primary-key presence, and FK out-degree. It provides a naming-independent proxy for the structural inventory that downstream artifacts need to describe. ColTypeF1 measures column-type composition, providing an aggregate proxy for the type requirements relevant to storage, transformations, and validation. These proxies distinguish dependency recovery, entity shape, and type composition, which can change independently. Correct join keys, entity correspondence, field-specific type compatibility, and agreement across artifacts require dedicated checks beyond these aggregate scores.

Table 1: Unified five-endpoint comparison (306 cases per condition). Bold marks the highest registered PlanGate score for each endpoint.

Condition	FKTopo				StructIso				ColTypeF1				SQLParse				DDLExec			
	SS	PG	Δ_p	Δ_a	SS	PG	Δ_p	Δ_a	SS	PG	Δ_p	Δ_a	SS	PG	Δ_p	Δ_a	SS	PG	Δ_p	Δ_a
DB-GPT / GPT-5.6 Sol	.711	–	.000		.603	–	.000		.739	–	.000		.712	–	-.049		.634	–	-.072	
DB-GPT / GPT-5.5	.396	–	-.315		.374	–	-.229		.553	–	-.186		.761	–	.000		.706	–	.000	
Data-Copilot / GPT-5.6 Sol	.450	–	-.262		.312	–	-.291		.609	–	-.130		.640	–	-.121		.314	–	-.392	
Data-Copilot / GPT-5.5	.390	–	-.322		.262	–	-.342		.622	–	-.117		.667	–	-.095		.294	–	-.412	
Qwen3-8B	.813	.832	+.019	+.121	.642	**.700**	+.058	+.097	.823	.873	+.050	+.134	.814	.912	+.098	+.150	.768	.869	+.101	+.163
Qwen3-32B	.788	.822	+.033	+.111	.648	.676	+.028	+.073	.853	.857	+.004	+.118	.873	**.928**	+.056	+.167	.830	**.892**	+.062	+.186
DeepSeek-V4-Flash	.786	**.859**	+.073	+.148	.616	.691	+.074	+.087	.804	**.885**	+.081	+.146	.817	.899	+.082	+.137	.768	.840	+.072	+.134
Qwen3-30B-A3B	.799	.814	+.014	+.102	.658	.667	+.009	+.064	.869	.872	+.003	+.133	.876	.882	+.006	+.121	.794	.807	+.013	+.101
MiniMax-m27	.708	.818	+.110	+.106	.556	.667	+.111	+.064	.806	.861	+.055	+.123	.853	.869	+.016	+.108	.788	.804	+.016	+.098
GPT-5.5	.547	.788	+.241	+.076	.432	.672	+.239	+.069	.815	.839	+.025	+.101	.915	.863	-.052	+.101	.732	.843	+.111	+.137
GPT-5.6 Sol	.595	.755	+.161	+.044	.535	.680	+.145	+.077	.805	.820	+.015	+.082	.948	.873	-.075	+.111	.824	.817	-.006	+.111

Table 1 reports five distinct outcomes. DeepSeek has the highest final FKTopo (0.8590); GPT-5.5 has the largest protocol gain (0.5466 to 0.7876, $\Delta_p = +0.2410$). GPT-5.6 Sol gains $+0.1609$. The three Qwen FKTopo intervals include zero. DeepSeek, MiniMax, and both GPT services have positive intervals and two-sided sign-flip $p < .001$. StructIso intervals are positive except for Qwen3-30B-A3B; ColTypeF1 intervals are positive only for Qwen3-8B, DeepSeek, and MiniMax. These unadjusted, model-stratified results do not establish simultaneous superiority. Appendix E provides estimates, uncertainty, and source-weighting sensitivity.

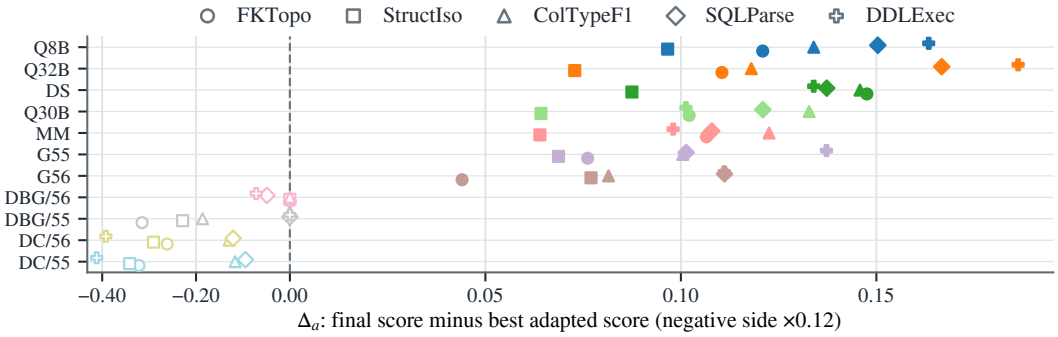

Figure 2: Final scores relative to the best adapted score for each metric (Δ_a) in one shared coordinate system. Color identifies the compact condition label on the y-axis; marker shape identifies the endpoint. Open markers are adapted protocols and filled markers are registered models. SQLParse and DDLExec use rate differences. The negative display region is visually compressed (ticks retain the original Δ_a values) so that the positive region remains readable. These descriptive contrasts do not use paired confidence intervals from the different within-model estimand Δ_p. *Abbreviations and notation for Figure 2 and other figures and tables.* SS and PG denote Single-shot and PlanGate; $\Delta_p = \mathrm{PG} - \mathrm{SS}$ is the paired protocol gain, and Δ_a is the final score minus the best adapted score for the same endpoint. Merged SS/PG cells contain adapted scores; "–" denotes an inapplicable protocol gain. Q8B/Q32B/Q30B denote Qwen3-8B/32B/30B-A3B, DS denotes DeepSeek-V4-Flash, MM denotes MiniMax-m27, and G55/G56 denote GPT-5.5/5.6 Sol. DBG and DC denote DB-GPT and Data-Copilot; /55 and /56 identify their GPT backends.

Relation recovery and reference composition. Qwen3-30B-A3B exceeds both GPT services on aggregate FKTopo but falls below them on the 224 declaration-bearing cases; DeepSeek remains highest in that stratum. Across 82 declaration-free cases, 8–31 outputs per model declare FKs yet resolve no scoring edge and receive FKTopo=1. We retain the registered 306-case aggregate and use the 224-case stratum as the primary relation-recovery sensitivity analysis. Appendix C reports unresolved predictions and signature-edge counts, which do not certify exact column-level FK correctness.

Execution and package verification. PlanGate improves SQLParse and original-script DDLExec for the five non-GPT services. GPT-5.5 improves execution despite lower parsing success; Sol improves neither. Complete-package checks remain a separate verification-coverage endpoint because transformations and tests are only partly supported (Appendix C.4). Only 2 transformation-rule and 132 validation-test artifacts are fully supported; a deterministic source-only renderer passes the strict check in 158/306 cases and builds normalized DDL in 236/306. It uses no reference answer, so these rates mix generated content with validator coverage.

Case variation and source composition. PlanGate reduces model-mean SD from 0.1000 to 0.0306 for FKTopo and from 0.0757 to 0.0114 for StructIso, but only 42–132 cases per model weakly dominate Single-shot across all three structural metrics. Weak dominance means no metric decreases and at least one increases. Appendix E reports the intervals, joint case outcomes, and source-weighting sensitivity.

5.2 Evidence channels and protocol mechanisms

The 2×2 design distinguishes dependence on the two tested channels from a protocol-specific change in how they interact. Figure 3A compares joint masking with fully visible input within each protocol (246 transformable cases). Single-shot FKTopo falls by 0.152–0.312 in the five non-GPT models, but rises by 0.1748 and 0.1667 in GPT-5.6 Sol and GPT-5.5; all seven contrasts survive the 196-hypothesis Holm correction. Under PlanGate, all seven point estimates are negative. The GPT-5.5 change is -0.2776 (adjusted $p = 0.01960$), whereas Sol's -0.0822 does not survive correction ($p = 0.47845$). Thus, channel removal changes structural recovery in a way that depends on both the model and generation protocol.

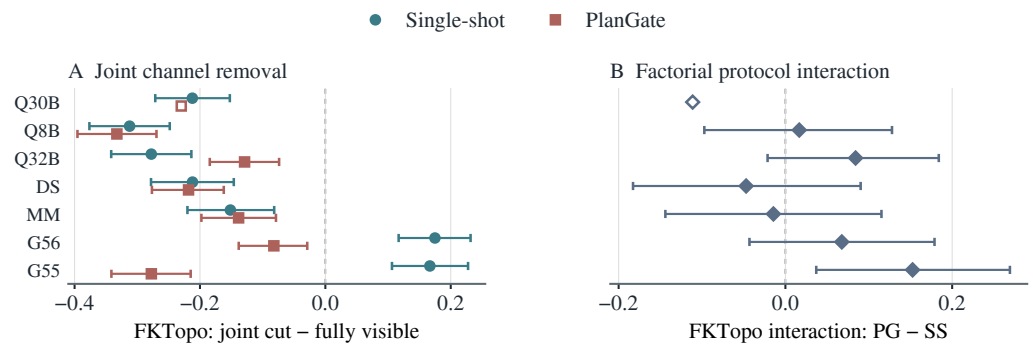

Figure 3: Key contrasts from the declaration-by-name intervention. (A) Joint masking minus fully visible input within each protocol. (B) PlanGate minus Single-shot in the factorial interaction, using the 168 cases supporting both transformations. Bars are unadjusted 95% case-bootstrap intervals; open points without bars are descriptive estimates. Table S6 gives exact values and corrected tests.

On the common factorial population, let $I_s = (Y_{s,11} - Y_{s,01}) - (Y_{s,10} - Y_{s,00})$ for protocol s. Figure 3B tests $I_{\mathrm{PG}} - I_{\mathrm{SS}}$; none of the complete comparisons survives correction. Even GPT-5.5's positive unadjusted interval, $[0.0370, 0.2693]$, has adjusted $p = 1$. The joint-removal effects therefore support protocol-dependent evidence use, without establishing that PlanGate changes declaration–name interaction. Table names and types remain visible; residual-context controls further reduce PlanGate FKTopo in six of seven models after correction (Appendix D), indicating useful evidence outside the two manipulated channels.

Generation versus selection. Matched streams locate corrected planning gains of $+0.274$ (GPT-5.6 Sol) and $+0.296$ (GPT-5.5), while gating changes the planned streams by -0.020 and -0.016; only planning survives the same Holm correction. The corrected gains therefore lie in candidate generation; gating and the plan–gate interaction do not reach that threshold (Appendix F).

6 Discussion and Limitations

The results position shared relational intent as an explicit interface between data-engineering design and delivery. PlanGate couples this interface to a reference-independent DDL predicate, making proposal and acceptance separately inspectable. The interventions show that declarations and naming cues affect structural recovery through model- and protocol-dependent paths; matched candidate streams locate the corrected GPT gains in planning. Structural fidelity and naming conformance should also be reported separately, because identifier overlap rewards naming conventions. Interpretation remains conditioned on the Spider-dominated source mix, single-reference scoring, and recorded service configurations. FKTopo can omit unresolved predictions, single-reference comparison can penalize valid alternatives, and execution is dialect-dependent. Consequently, structural recovery, executable DDL, and package verification are distinct delivery outcomes; reporting each alongside its checking coverage connects model evaluation to the requirements of modeling, transformation, loading, and validation.

7 Conclusion

PlanGate provides a shared relation context and a deterministic DDL submission boundary for the complete eight-artifact task. On the 306-case DE-VeriBench corpus, mean FKTopo is higher than Single-shot for all seven models, while uncertainty and other endpoints vary by model. Evidence interventions reveal sensitivity that depends on the model and protocol. Matched-stream controls locate corrected gains for both GPT models in planning-induced generation; gating and the plan–gate interaction do not meet the correction threshold. Package verification remains a distinct outcome with incomplete checking coverage.

For data engineering, shared relation plans express intended dependencies across modeling, transformation, loading, and validation. Their preservation should be assessed separately from execution and cross-artifact agreement, including future tests during schema evolution and maintenance.

AI USE STATEMENT

Generative AI tools assisted manuscript writing and language polishing, literature retrieval and discovery, and the natural-language realization of a limited synthetic portion of the dataset. Dataset constraints, measurements, and reported numerical results were determined by deterministic rules, executable checks, recorded experiment data, and author-controlled scripts. The authors are responsible for verifying the manuscript, citations, data description, and supporting artifacts.

REFERENCES

Samuel R. Bowman and George E. Dahl. What will it take to fix benchmarking in natural language understanding? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4843–4855, 2021. URL https://aclanthology.org/2021.naacl-main.385/.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6s5uXNWGIh.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=KuPixIqPiq.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005. URL https://research.ibm.com/publications/data-exchange-semantics-and-query-answering.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=XmProj9cPs.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C.-C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=dI4wzAE6uV.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research (TMLR)*, 2023. URL https://openreview.net/forum?id=iO4LZibEqW.

Chunwei Liu, Matthew Russo, Michael Cafarella, Lei Cao, Peter Baile Chen, Zui Chen, Michael Franklin, Tim Kraska, Samuel Madden, Rana Shahout, and Gerardo Vitagliano. Palimpzest: Optimizing ai-powered analytics with declarative query processing. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2025a. URL https://vldb.org/cidrdb/papers/2025/p12-liu.pdf.

Yurong Liu, Eduardo H. M. Pena, Aécio Santos, Eden Wu, and Juliana Freire. Magneto: Combining small and large language models for schema matching. *Proceedings of the VLDB Endowment*, 18(8):2681–2694, 2025b. URL https://www.vldb.org/pvldb/vol18/p2681-freire.pdf.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

Ansong Ni, Srini Iyer, Dragomir Radev, Ves Stoyanov, Wen-tau Yih, Sida I. Wang, and Xi Victoria Lin. Lever: Learning to verify language-to-code generation with execution. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 202, pp. 26106–26128. PMLR, 2023. URL https://proceedings.mlr.press/v202/ni23b.html.

Liana Patel, Siddharth Jha, Melissa Pan, Harshit Gupta, Parth Asawa, Carlos Guestrin, and Matei Zaharia. Semantic operators and their optimization: Enabling LLM-based data processing with accuracy guarantees in LOTUS. *Proceedings of the VLDB Endowment*, 18(11):4171–4184, 2025. URL https://www.vldb.org/pvldb/vol18/p4171-patel.pdf.

Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data quality assessment. *Communications of the ACM*, 45 (4):211–218, 2002. doi: 10.1145/505248.506010.

Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. Synchromesh: Reliable code generation from pre-trained language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=KmtVD97J43e.

Mohammadreza Pourreza, Hailong Li, Ruoxi Sun, Yeounoh Chung, Shayan Talaei, Gaurav Tarlok Kakkar, Yu Gan, Amin Saberi, Fatma Ozcan, and Sercan O. Arik. CHASE-SQL: Multi-path reasoning and preference optimized candidate selection in text-to-sql. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=CvGqMD5OtX.

Mushtari Sadia, Zhenning Yang, Yunming Xiao, Ang Chen, and Amrita Roy Chowdhury. SQUiD: Synthesizing relational databases from unstructured text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 31987–32012, 2025. URL https://aclanthology.org/2025.emnlp-main.1629/.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. URL https://aclanthology.org/2021.emnlp-main.779/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Sanjit Seshia, and Vijay Saraswat. Combinatorial sketching for finite programs. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 404–415. ACM, 2006. doi: 10.1145/1168857.1168907.

Emina Torlak and Rastislav Bodík. A lightweight symbolic virtual machine for solver-aided host languages. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pp. 530–541. ACM, 2014. doi: 10.1145/2594291.2594340.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023. URL https://aclanthology.org/2023.acl-long.147/.

Qin Wang, Youhuan Li, Yansong Feng, Si Chen, Ziming Li, Pan Zhang, Zihui Si, Yixuan Chen, Zhichao Shi, Zebin Huang, Guo Chen, and Wenqiang Jin. Text2Schema: Filling the gap in designing database table structures based on natural language, 2025. URL https://arxiv.org/abs/2503.23886v2. Version 2.

Brandon T. Willard and Rémi Louf. Efficient guided generation for large language models, 2023. URL https://arxiv.org/abs/2307.09702.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen llm applications via multi-agent conversations. In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Siqiao Xue, Yunjia Fan, Xiaoyan Zhang, et al. Db-gpt: Empowering database interactions with private large language models, 2023. URL https://arxiv.org/abs/2312.17449.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. URL https://aclanthology.org/D18-1425/.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=z5uVAKwmjf.

Tianshu Zhang, Kun Qian, Siddhartha Sahai, Yuan Tian, Shaddy Garg, Huan Sun, and Yunyao Li. Evoschema: Towards text-to-sql robustness against schema evolution. *Proceedings of the VLDB Endowment*, 18(10): 3655–3668, 2025b. URL https://www.vldb.org/pvldb/vol18/p3655-zhang.pdf.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Data-copilot: Bridging billions of data and humans with autonomous workflow, 2023. URL https://arxiv.org/abs/2306.07209.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

This supplement provides the generation protocol, benchmark composition, metric definitions, and complete experimental comparisons. Each comparison identifies its models, case population, and statistical unit. Registered tests and subsequent sensitivity analyses are reported separately.

For orientation, the main protocol is a single path from visible source schema to a shared relation plan, joint realization of the eight-artifact Ω_8 package, deterministic DDL admission, and independent scoring. The appendix expands each interface without changing the reported task: the four adapted endpoint-calibration conditions are DB-GPT and Data-Copilot paired with GPT-5.6 Sol and GPT-5.5, all evaluated on the same DE-VeriBench cases and frozen evaluator.

The full-corpus comparison and expanded interventions use distinct support rules. All seven models have complete 306-case coverage in both structural-comparison protocols. Model labels identify recorded serving configurations; their shared case membership supports aligned outcome comparisons, while service differences preclude a common-budget capability ranking. Specialized mechanism and operational experiments retain their actual model scope. The same case can occur in several experiments, so the entries must not be added to obtain a count of independent samples.

A Task Contract, Notation, and Protocol

A.1 Inputs, outputs, and the relation plan

A case is $x = (S, R, \Gamma, X, \Omega)$: source schema or source metadata S, requested requirements R, governance constraints Γ, execution context X, and the requested artifact contract Ω. The evaluated contract Ω_8 requires the ordered package

$$P_{\Omega_8} = (v_{\mathrm{LM}}, v_{\mathrm{DDL}}, v_{\mathrm{ERD}}, v_{\mathrm{Dict}}, v_{\mathrm{Map}}, v_{\mathrm{Transform}}, v_{\mathrm{DAG}}, v_{\mathrm{Test}}).$$

The subscripts denote a logical model, schema DDL, entity–relationship view, data dictionary, source mapping, transformation rules, load DAG, and validation tests. A relation plan Q represents intended directed foreign-key tuples (t_s, c_s, t_t, c_t), with source table and column (t_s, c_s) and target table and column (t_t, c_t). The same Q conditions all eight views through its consumed prefix $\widetilde{Q}$, held fixed across retries. The minimal plan requests FK relations only; the full-plan sensitivity additionally requests tables and primary keys. In the live implementation, the planner receives source-schema text and produces a free-text FK list. Three quantities must be kept separate: $L = 400$ is the character prefix consumed by the generator; 400 output tokens is the planner's nominal request; and the instrumented full-corpus transport raises that request budget to at least 12,000 output tokens for the recorded service configuration. The 12,000-token value is a transport/request ceiling, not an observed per-case output count and not a character count. It does not change L or guarantee that the retained prefix covers every relation. These prompt conditions are not validated, canonical graph objects. The task tuple describes a conceptual interface. In the reported experiments, the model sees case-specific source-schema text and a fixed eight-artifact system prompt; the other tuple fields are not independently supplied inputs. For the 65 dbt cases, the input is the model SQL/source text supplied with the case rather than a complete source DDL; missing source-table statistics do not mean that the model receives an empty input.

We distinguish the proposed plan from the injected context as $\widetilde{Q} = T_L(Q) = Q[1 : \min(L, |Q|)]$, where L counts characters, not output tokens. The primary setting is $L = 400$. The expanded budget comparison holds the planning output fixed and injects 400 characters, 1,200 characters, or the complete text. It does not adapt the budget to each case. The full source schema remains visible to generation. A common character limit fixes the added text allowance, not the fraction of relations retained, and has not been established as optimal. Character truncation can split identifiers and does not guarantee complete relation coverage. The registered endpoint for the length comparison is strict package pass. All complete paired comparisons have zero differences, intervals $[0, 0]$, and adjusted $p = 1$. This endpoint does not distinguish the length settings; structural and semantic quality require their own outcome measures.

A.2 Relation-plan transport audit

The protocol retains the first 400 characters of the planner's free-text relation plan. In the seven-model comparison, all 294 nonempty recorded Qwen3-30B-A3B plan responses exceed this prefix

budget (median length 1,969 characters). This measures text retention; relation coverage additionally depends on the order and representation of relations in the consumed prefix.

For candidate v_b at realization index b, the reported minimal acceptance predicate is

$$A_1(v_b) = \mathbf{1}[\mathrm{parse}(v_{b,\mathrm{DDL}}) = 1]\mathbf{1}[\mathrm{refs}(v_{b,\mathrm{DDL}}) \geq 1].$$

The parser is the frozen multi-dialect parser used by the experiment and success means that it recovers at least one table. The predicate reads neither the reference package nor any learned score. It is therefore a commit rule, not a quality oracle. The live predicate counts the case-sensitive literal `REFERENCES`, rather than traversing foreign-key AST nodes. Passing A_1 establishes only a parser-recognized table and a minimum lexical reference count; it does not certify reference resolution, database execution, lineage consistency, or eight-artifact package correctness.

The main comparison uses the minimal $\tau = 1$ predicate, fixed by protocol before outcome analysis.

A.3 Deployment and strict semantics

PlanGate protocol. (1) Generate one compact relation plan Q from x. (2) For $b = 1, \ldots, B$ with $B = 3$, realize all eight artifacts under the same consumed prefix $\widetilde{Q}$ and evaluate $A_1(v_b)$. (3) Return the first passing candidate. (4) If none passes, the *fallback* estimand applies the track-specific return rule below, whereas the *strict* estimand abstains.

The main full-corpus structural comparison uses deployment fallback and scores all 306 paired cases. Strict success, accepted-subset quality, original-script SQLite execution, and measured package consistency are separate endpoints. The seven-model main comparison returns the final candidate slot when the generation budget is exhausted. Both service interfaces apply this return rule. Every registered case remains in the scoring denominator, and an empty candidate follows the registered scoring convention. Fallback and strict abstention therefore define separate deployment outcomes.

A.4 Intervention notation

The registered intervention independently manipulates two relation-evidence channels: $D = 1$ retains explicit foreign-key declarations and $N = 1$ retains meaningful column names. $D = 0$ removes declarations; $N = 0$ replaces column identifiers with opaque names. Thus $(D, N) \in \{0, 1\}^2$. Table names are not manipulated, so the joint-cut cell removes the two tested channels rather than all conceivable evidence. The design is a 2×2 factorial because both interventions are binary; an additional 0/25/50/75/100% masking experiment measures continuous declaration intensity.

Masking precedes planning: both the planner and generator receive the same already-masked source text. Opaque identifiers remain opaque throughout generation. The inverse map is built from source-column renaming and applied only to generated DDL for scoring; it is not sent to either model stage and does not derive from the reference graph. Raw/inverse-mapped score parity is checked separately. A fixed plan conditions generation; the gate does not test whether its individual edges appear in any output view.

Conditional identifiability and the intervention's scope. If two different target relation graphs yield exactly the same visible input, a fixed procedure without external information has the same output distribution for both and cannot guarantee exact recovery of both targets. The registered intervention does not establish this premise: table names, types, and model priors remain available. Its four cells distinguish the conditional contributions of declarations and meaningful column names, including whether one channel compensates for removal of the other. The empirical result is not an information-theoretic impossibility claim.

A.5 Detailed generation and acceptance workflow

The step-by-step view is summarized by the main overview. The FK-only plan, $\tau = 1$, and $B = 3$ define the reported minimal protocol. Plan-scope and threshold alternatives are sensitivity settings, not changes to the reported default.

Figure S1 illustrates why representing a relation in one view does not ensure that it appears in the DDL. The first candidate has an ERD edge but no declaration witness. Rejection triggers another generation under the same plan; it does not supply a repair instruction.

Fixed relation plan. `orders.customer_id -> customers.id`
Candidate 1. The ERD includes this dependency. The DDL column is `customer_id INTEGER`, with no reference declaration. The parser recovers tables, but the literal reference count is zero: *reject and regenerate under the same plan*.
Candidate 2. The DDL column is `customer_id INTEGER REFERENCES customers(id)`. If parsing succeeds, the minimal predicate passes: *return the admitted package*.
Independent evaluation. Check whether the diagram, dictionary, mapping, load order, and tests agree with the DDL. These checks do not participate in the admission predicate.

Figure S1: Illustrative two-table generation and admission, not an observed experimental trace. The fixed plan links `orders.customer_id` to `customers.id`. Candidate 1 depicts the relation in its ERD but omits `REFERENCES` in the DDL and is rejected. A subsequent candidate adds the declaration and passes. The gate checks parsing and lexical declaration count, not each planned edge. Other views and executable constraints are evaluated independently. DDL fragments show the relevant column definitions; the complete tables and remaining artifacts are omitted for readability.

The live planner's system instruction is `List FK relationships: child.col -> parent.col`; its user message contains the visible source schema. The generator's fixed eight-view system instruction is extended with `Include these FKs:` followed by the locked plan prefix, and its user message again contains that same source schema. For example, the plan tuple `orders.customer_id -> customers.id` asks the generator to realize a DDL reference, a corresponding ERD dependency, consistent dictionary key descriptions, a source-to-target key mapping, transformations that preserve those keys, a parent-before-child load dependency, and dependency tests, alongside the logical model. These are intended correspondences induced by one shared prompt condition, not independently enforced constraints. The generator requests a JSON object with key `views`, containing `logical_model`, `ddl`, `erd`, `data_dictionary`, `source_mapping`, `transformation_rules`, `load_dag`, and `validation_tests`, each represented as an array of strings. Its instruction specifies `ddl = CREATE TABLE with FK constraints. Output ONLY JSON.` Rejection regenerates the package under the same source and plan without an additional repair prompt.

Architecture example and display conventions. The main architecture uses a public MIMIC-derived seven-table schema sketch: PATIENTS, ADMISSIONS, ICUSTAYS, CHARTEVENTS, LABEVENTS, DIAGNOSES_ICD, and NOTEEVENTS. It supplies no clinical records and declares no foreign keys. SUBJECT_ID, HADM_ID, and ICUSTAY_ID provide patient, admission, and ICU-stay context; displayed relations and attempt traces illustrate the mechanism, not observed experimental trajectories. The evidence-probe inset illustrates four cue states; the actual interventions operate case-wide before planning. Fallback and admission are separate protocol events; relation-free inputs are handled by the registered return rule. Gold never enters generation or admission; returned outputs remain in scoring. When the generation budget is exhausted without admission, the protocol returns the final candidate. All seven deployed models have complete 306-case structural comparisons (Appendix E).

The gate provides only the DDL acceptance test defined above; it neither repairs a missing correspondence nor verifies the meaning of a field description.

B Benchmark Provenance and Audit

B.1 Composition

The benchmark contains 306 cases from seven source packages. Its development, test, and stress labels describe case organization: all 306 cases are evaluated in the full-corpus comparison, and none is used to fit model parameters.

Table S1: Benchmark composition by source and task role.

Source package	Cases	Share	Principal task role
Spider	166	54.2%	schema reconstruction
Velir dbt-ga4	60	19.6%	reverse engineering
TPC-derived	36	11.8%	warehouse stress cases
MIMIC schema/governance	18	5.9%	governance-aware design
BIRD	11	3.6%	schema reconstruction
Synthetic scaffold	10	3.3%	controlled edge cases
dbt jaffle-shop	5	1.6%	reverse engineering/mapping
Total	306	100%	183 dev / 92 test / 31 stress

B.2 Schema size and recorded domain labels

Size refers to schema declarations, not stored records. Complete source DDL is available for 241 cases; the remaining 65 dbt cases contain model SQL, so their complete source-table and source-column counts are unavailable rather than zero. For these dbt cases, the model receives the recorded model-SQL/source-schema text supplied by the case package; unavailable counts mean that complete source DDL was not available, not that the source schema is empty. The 241 complete source schemas contain 1,220 table and 6,608 column occurrences. All 306 target reference schemas together contain 1,285 table and 7,010 column occurrences. Counts sum across cases, rather than deduplicating identically named objects from different tasks.

Table S2: Cumulative schema dimensions by source package. Source counts cover complete DDL inputs; "–" denotes model-SQL input without complete source DDL.

Source	Cases	Source tables	Source columns	Target tables	Target columns
Spider	166	876	4,503	876	4,503
dbt-GA4	60	–	–	60	375
TPC-derived	36	145	742	145	742
MIMIC schema/governance	18	84	445	84	445
BIRD	11	75	798	75	798
Synthetic scaffold	10	40	120	40	120
dbt jaffle-shop	5	–	–	5	27
Total	306	1,220	6,608	1,285	7,010

The source manifests contain eleven recorded domain-label values, but these mix application labels with source-package and design-setting labels; they do not establish eleven independently curated application domains.

Table S3: Recorded domain and design-setting groups, with cumulative reference dimensions, artifact requirements, and validation assertions.

Recorded group	Cases	Tables	Columns	Artifacts	Assertions
Analytics	65	65	402	520	49
BIRD source schemas	11	75	798	88	126
Clinical schema/governance	18	84	445	144	314
Decision-support warehouse	18	64	325	144	209
E-commerce	2	8	24	16	38
Synthetic finance	2	8	24	16	38
Oil/gas master data	2	8	24	16	38
Organization	2	8	24	16	38
Retail warehouse	18	81	417	144	249
Spider source schemas	166	876	4,503	1,328	2,524
Supply chain	2	8	24	16	38
Total	306	1,285	7,010	2,448	3,661

Each of the eight artifact categories is requested in all 306 cases, giving 2,448 artifact requirements. The reference validation files contain 3,661 assertions. These totals describe the task contract and reference checks; generated-package coverage and executed tests are evaluated separately.

B.3 Approval, duplication, and contamination boundaries

All 306 derived case records underwent automated construction and model-agent checking of the derived case package. Reference artifacts retain their source-benchmark provenance. Exact normal-

ized source-schema hashing identifies five duplicate pairs; reference-DDL hashing finds nine groups, comprising eight pairs and one triplet and including all five source pairs. Three distinct pairs cross nominal split labels, so these labels are not content-disjoint. The union yields 296 clusters: 287 singletons, eight pairs, and one triplet. This audit identifies a potential weighting sensitivity; the current seven-model estimates use the same 306 case weights. This exact-duplicate analysis measures repeated case weighting; near-duplicate similarity and pretraining exposure remain separate sources of uncertainty.

B.4 Release policy and licenses

Table S4: Release action by source; "recipe only" exposes identifiers and hashes without source payload.

Source	Terms / review	Anonymous release action
Spider	CC-BY-SA-4.0; verified	reviewed metadata or fetch-and-verify recipe with attribution
Velir dbt-ga4	MIT; verified	derived metadata with pinned upstream notice
TPC-derived	specification terms; open review	recipe only; no kit or specification payload
MIMIC-derived	credentialed boundary; open review	public-document recipe only; no records or credentialed files
BIRD	CC-BY-SA-4.0; verified	reviewed metadata or fetch-and-verify recipe with attribution
Synthetic	project controlled; verified	generated identifiers and rules only
dbt jaffle-shop	Apache-2.0; verified	reviewed metadata with notice

TPC- and MIMIC-derived source content is excluded until per-file rights review is closed. These restrictions limit data release, not the aggregate result audit.

C Metrics, Calibration, and Execution Endpoints

C.1 Evaluation metrics

Evaluation separates structural fidelity, naming agreement, execution, and cross-artifact consistency. SQLParse is the fraction of cases whose emitted DDL the SQLGlot evaluator parses into at least one table. DDLExec (SQLite) requires the emitted script to execute in a fresh in-memory SQLite database with foreign-key checking enabled. The package evaluator separately reports schema-build success after SQLGlot normalization to SQLite; this translation step changes dialect compatibility and is identified explicitly in its results. Both protocols within a comparison use the same evaluator and denominator. Package evaluation then measures agreement between the schema, dictionary, relation views, mappings, load dependencies, and executable tests.

For the main comparison, FKTopo is naming-independent recovery of resolved foreign-key edges, StructIso is F1 over multisets of table signatures, and ColTypeF1 is F1 over multisets of canonical column types. SQLParse measures syntactic parseability, while DDLExec measures direct execution of the original script in SQLite with foreign-key checks enabled. The five endpoints therefore separate structural fidelity, parseability, and direct execution.

For each parsed table t, the frozen scorer forms a signature $s(t) = (n_t, C_t, k_t, d_t)$: column count n_t, sorted canonical column types C_t, primary-key presence k_t, and the number d_t of parsed outgoing foreign-key declarations. Let $\mathcal{T}$, $\mathcal{E}$, and $\mathcal{C}$ be the multisets of table signatures, signature pairs for resolved directed FK edges, and column types, respectively. A star denotes the reference and a hat the generated DDL. The three primary metrics are:

$$\text{FKTopo} = F_1(\mathcal{E}^*, \widehat{\mathcal{E}}),$$
$$\text{StructIso} = F_1(\mathcal{T}^*, \widehat{\mathcal{T}}),$$
$$\text{ColTypeF1} = F_1(\mathcal{C}^*, \widehat{\mathcal{C}}).$$

For nonempty total cardinality, multiset F1 is twice the intersection cardinality divided by the sum of reference and predicted cardinalities. It equals one when both multisets are empty and zero when only one is empty. Scores are computed per case and macro-averaged. The experiment wrapper assigns its registered empty-return convention; this is distinct from the scorer's empty-multiset convention.

The scorer performs no optimal table matching: StructIso is not an exact graph-isomorphism test, and distinct tables can share a signature. An FK edge enters $\mathcal{E}$ only if both table endpoints have local

signatures; unresolved references require a separate diagnostic. SchemaF1, called RAWSCHEMAF1 in the calibration, is table-name set F1; FieldF1 is F1 between reference field names and generated data-dictionary field names, represented as table–column pairs. Both remain naming-sensitive diagnostics rather than structural outcomes.

C.2 COMPLETE SEVEN-MODEL REFERENCE AND SCORE AUDIT

Reference DDL defines 82 cases without FK declarations and 224 with declarations. Two declaration-bearing references contain only unresolved references, giving 84 empty and 222 nonempty resolved reference graphs. All seven models use these same strata and case denominators. The analysis uses saved outputs whose FKTopo values were reproduced with the unchanged frozen evaluator.

Table S5: Seven-model scoring diagnostics. Full/partial/zero FKTopo counts use all 224 FK-declared cases. Edge-signature matches use the same 224 cases and 1,049 reference signature edges; they do not certify exact FK correctness. Blind cases declare FKs but have an empty scored graph and score 1, with all 82 declaration-free references as the denominator.

Model	SS: 1/p/0	PG: 1/p/0	Match/pred.	Match/ref.	Blind/82
Q30B	152/36/36	153/38/33	779/984	779/1049	26/82
Q8B	156/33/35	160/43/21	819/989	819/1049	24/82
Q32B	147/33/44	161/40/23	837/1048	837/1049	26/82
DS	145/33/46	176/27/21	851/962	851/1049	31/82
MM	120/41/63	156/45/23	805/1044	805/1049	26/82
G56	105/50/69	159/39/26	795/987	795/1049	8/82
G55	83/42/99	162/34/28	820/1069	820/1049	9/82

On declaration-bearing references, PlanGate FKTopo in the fixed order Qwen3-30B-A3B, Qwen3-8B, Qwen3-32B, DeepSeek-V4-Flash, MiniMax-m27, GPT-5.6 Sol, GPT-5.5 is 0.7854, 0.8290, 0.8237, 0.8564, 0.8182, 0.8132, and 0.8169. On declaration-free references the means are 0.8902, 0.8415, 0.8171, 0.8659, 0.8171, 0.5976, and 0.7073. DeepSeek has the highest all-case and declaration-bearing means; Qwen3-30B-A3B exceeds both GPT services overall but not on declaration-bearing references. These post-hoc descriptive comparisons characterize scoring sensitivity, without identifying intrinsic model ability or a complete causal explanation of score differences.

On the 224 declaration-bearing cases, perfect/partial/zero PlanGate FKTopo counts are respectively 153/38/33, 160/43/21, 161/40/23, 176/27/21, 156/45/23, 159/39/26, and 162/34/28 in the same model order. The corresponding Single-shot counts are 152/36/36, 156/33/35, 147/33/44, 145/33/46, 120/41/63, 105/50/69, and 83/42/99. Each triplet sums to 224; perfect score means signature overlap, not verified table or column identities.

The same 224 references contain 1,049 resolved signature edges. Within-case multiset intersections yield matched/predicted edge counts of 779/984, 819/989, 837/1048, 851/962, 805/1044, 795/987, and 820/1069. The common reference denominator permits a direct edge-level comparison, but these counts measure name-independent endpoint-signature matches. They are neither the macro-average FKTopo nor a strict FK-declaration correctness rate. Unresolved declarations do not enter the predicted-edge denominator.

This omission interacts with empty/empty scoring. Of the same 82 reference-declaration-free cases, 26, 24, 26, 31, 26, 8, and 9 PlanGate outputs respectively contain predicted FK declarations but resolve no scored edge and receive FKTopo=1. These complete-denominator counts expose a blind spot across all seven models. A resolved added relation is penalized against an empty reference graph, whereas an unresolved added relation can disappear from the scored graph. Reference composition and edge resolution can therefore alter aggregate model order under the unchanged scorer.

C.3 CONTROLLED METRIC CALIBRATION

Reference-side corruption checks use the applicable subsets of the 306 references. FK deletion uses 224 references with declarations, and table deletion uses 239 multi-table references. Dangling-reference injection, type flips, known-prefix renaming, arbitrary renaming, and dependency-cycle

injection use all 306 references. All unmodified references have self-scores of 1.000 under the frozen evaluator.

Complete FK deletion lowers FKTOPO from 1.000 to 0.009 on the 224-reference subset, while COLTYPEF1 remains 1.000. The residual is exactly 2/224: two references contain declarations but no resolved internal edges, retaining an empty/empty score of 1. A post-hoc sensitivity analysis restricted to the 222 nonempty resolved reference graphs gives a complete-deletion FKTOPO of 0.000. This endpoint removes every matching declaration line and does not sample intermediate corruption levels.

Arbitrary renaming leaves STRUCTISO at 1.000 while RAWSCHEMAF1 falls to 0.355. These checks establish targeted sensitivity and naming invariance, without establishing complete construct validity or semantic equivalence to every valid alternative.

C.4 CURRENT PACKAGE VERIFICATION AND REPRESENTATION COVERAGE

Strict package success requires all eight requested artifacts and every applicable consistency and execution check to pass. The evaluator parses supported representations into typed structures and checks column-level relationships, dictionary/schema agreement, mappings, load dependencies, and recognized validation tests. Partially supported descriptions remain incompletely verifiable, even when all recognized checks pass. The current adapters provide incomplete coverage for most generated packages, so this strict endpoint primarily measures demonstrated automatic verification coverage. Only 2 transformation-rule artifacts and 132 validation-test artifacts are fully supported by the adapters. These counts describe verification coverage, not the fraction of semantically incorrect designs.

The package evaluator renders DDL through SQLGlot and builds isolated in-memory SQLite databases. Bounded synthetic rows support constraint probes and recognized read-only validation queries. This normalized build is separate from the original-script DDLExec endpoint in the main comparison. It does not establish native-dialect equivalence or production behavior.

A deterministic source-only eight-artifact renderer passes the strict check in 158/306 cases and builds DDL in 236/306 cases. Its machine-readable format has a direct adapter advantage; these results provide a validator-specific reference, not semantic superiority over free-form outputs. The renderer sees no reference answer. Each source-only output is reused across models rather than treated as an independent realization for every comparison.

None of the 1,224 adapted DB-GPT and Data-Copilot packages passes the strict all-artifact checks. The expanded component campaign is a separate population: its unplanned-first and plan-plus-gate conditions contain 2,142 and 2,141 model–case outputs, respectively, under the same strict adapter boundary. The missing planned output belongs to Qwen3-30B-A3B. These are outcomes under different conditions, not additional independent benchmark cases. The component comparisons in Appendix F report structural effects separately.

D REGISTERED CHANNEL INTERVENTION

D.1 EXPANDED INTERVENTION AND DOSE ANALYSIS

The expanded campaign registers 306 cases per model. Declaration removal is operable in 168 cases, while column-name and joint masking are supported in 246 cases. The factorial protocol interaction uses the 168 cases with both operable channels; joint-removal contrasts use the 246 transformable cases. Five-level dose analysis uses actual deletion fractions and deduplicates unchanged transformed inputs before estimating within-case slopes. For m parsed visible declarations and nominal deletion percentage $p \in \{0, 25, 50, 75, 100\}$, the implementation removes $k = \lfloor (pm + 50)/100 \rfloor$ declarations (nearest integer, with halves rounded up). The first k entries of a case- and seed-specific shuffled declaration order are removed, making the doses nested. The actual fraction is recomputed from parsed counts before and after transformation; identical transformed-input hashes are aliases rather than independent observations.

The following tables retain all point estimates. Inferential intervals and adjusted tests are reported for complete paired comparisons; marked subset estimates are descriptive. All 196 hypotheses remain in the Holm family, and exact coverage records are provided in the analysis package.

Table S6: Final-output comparisons. Intervals are unadjusted case-bootstrap 95% CIs; p values use sign flips and the full 196-hypothesis Holm family. Joint removal subtracts fully visible input within protocol. Factorial protocol DiD is PlanGate minus Single-shot in the declaration-by-name interaction on the dual-operable population. All complete factorial contrasts have adjusted $p = 1$, including GPT-5.5.

Contrast	Estimate	Unadjusted 95% CI	$p_{\mathrm{Holm}(196)}$
Q30B			
Joint cut: Single-shot	-0.2122	[-0.2713, -0.1524]	0.01960
Joint cut: PlanGate	-0.2299	–	–
Factorial protocol DiD	-0.1112	–	–
Q8B			
Joint cut: Single-shot	-0.3122	[-0.3764, -0.2482]	0.01960
Joint cut: PlanGate	-0.3326	[-0.3954, -0.2694]	0.01960
Factorial protocol DiD	+0.0166	[-0.0972, +0.1278]	1.00000
Q32B			
Joint cut: Single-shot	-0.2776	[-0.3416, -0.2137]	0.01960
Joint cut: PlanGate	-0.1289	[-0.1844, -0.0738]	0.01960
Factorial protocol DiD	+0.0842	[-0.0214, +0.1839]	1.00000
DS			
Joint cut: Single-shot	-0.2119	[-0.2781, -0.1461]	0.01960
Joint cut: PlanGate	-0.2182	[-0.2766, -0.1620]	0.01960
Factorial protocol DiD	-0.0471	[-0.1827, +0.0902]	1.00000
MM			
Joint cut: Single-shot	-0.1515	[-0.2200, -0.0816]	0.01960
Joint cut: PlanGate	-0.1384	[-0.1977, -0.0788]	0.01960
Factorial protocol DiD	-0.0144	[-0.1440, +0.1152]	1.00000
G56			
Joint cut: Single-shot	+0.1748	[+0.1169, +0.2316]	0.01960
Joint cut: PlanGate	-0.0822	[-0.1382, -0.0289]	0.47845
Factorial protocol DiD	+0.0676	[-0.0431, +0.1788]	1.00000
G55			
Joint cut: Single-shot	+0.1667	[+0.1063, +0.2276]	0.01960
Joint cut: PlanGate	-0.2776	[-0.3413, -0.2148]	0.01960
Factorial protocol DiD	+0.1525	[+0.0370, +0.2693]	1.00000

Table S7: Final-output comparisons. Intervals are unadjusted case-bootstrap 95% CIs; p values use sign flips and the full 196-hypothesis Holm family. Within-case FKTopo slopes use actual removal fractions with identical inputs deduplicated. The difference row averages paired case-level slope differences.

Contrast	Estimate	Unadjusted 95% CI	$p_{\mathrm{Holm}(196)}$
Q30B			
Single-shot slope	-0.1265	[-0.1785, -0.0771]	0.01960
PlanGate slope	-0.1412	–	–
PlanGate minus Single-shot	-0.0124	–	–
Q8B			
Single-shot slope	-0.2451	[-0.3089, -0.1825]	0.01960
PlanGate slope	-0.2929	[-0.3506, -0.2358]	0.01960
PlanGate minus Single-shot	-0.0478	[-0.1216, +0.0258]	1.00000
Q32B			
Single-shot slope	-0.1242	[-0.1741, -0.0761]	0.01960
PlanGate slope	-0.1261	[-0.1844, -0.0687]	0.01960
PlanGate minus Single-shot	-0.0018	[-0.0572, +0.0539]	1.00000
DS			
Single-shot slope	-0.1259	[-0.1977, -0.0548]	0.13359
PlanGate slope	-0.1596	[-0.2174, -0.1014]	0.01960
PlanGate minus Single-shot	-0.0336	[-0.1242, +0.0560]	1.00000
MM			
Single-shot slope	-0.0249	[-0.1002, +0.0507]	1.00000
PlanGate slope	-0.0555	[-0.1110, -0.0005]	1.00000
PlanGate minus Single-shot	-0.0306	[-0.1125, +0.0512]	1.00000
G56			
Single-shot slope	+0.0429	[-0.0212, +0.1080]	1.00000
PlanGate slope	-0.0767	[-0.1335, -0.0205]	1.00000
PlanGate minus Single-shot	-0.1197	[-0.2011, -0.0374]	0.71713
G55			
Single-shot slope	+0.1722	[+0.0995, +0.2463]	0.01960
PlanGate slope	-0.3184	[-0.3900, -0.2470]	0.01960
PlanGate minus Single-shot	-0.4906	[-0.5897, -0.3944]	0.01960

What remains after joint masking? The residual-context control further anonymizes table and constraint names and removes comments while preserving types and the remaining schema structure. It asks whether the joint-mask output still depends on these additional cues, rather than assuming that two removed channels exhaust the available evidence. Table S8 reports both protocols for every model under the same 246-case support conditions rule. These contrasts measure an additional input intervention, not the ability of a verifier to reconstruct hidden information. All seven gated estimates are negative, with six remaining significant after Holm correction (Qwen3-8B is the exception). No Single-shot contrast reaches that threshold. The result shows that useful context remains after the first two channels are removed, particularly for the planned protocol; significance in one protocol but not the other is not itself a test of their difference.

Table S8: Final-output residual-context control: anonymize table and constraint names and remove comments after joint masking. Entries are residual minus joint-mask FKTopo. Intervals are unadjusted; p uses the full 196-comparison Holm family.

Contrast	Estimate	Unadjusted 95% CI	$p_{\mathrm{Holm}(196)}$
Q30B			
Residual cut: Single-shot	-0.0429	[-0.1023, +0.0176]	1.00000
Residual cut: PlanGate	-0.1616	[-0.2222, -0.1018]	0.01960
Q8B			
Residual cut: Single-shot	-0.0897	[-0.1577, -0.0208]	1.00000
Residual cut: PlanGate	-0.1030	[-0.1653, -0.0396]	0.21578
Q32B			
Residual cut: Single-shot	+0.0040	[-0.0626, +0.0692]	1.00000
Residual cut: PlanGate	-0.1466	[-0.2088, -0.0849]	0.01960
DS			
Residual cut: Single-shot	-0.0432	[-0.1152, +0.0296]	1.00000
Residual cut: PlanGate	-0.1999	[-0.2680, -0.1305]	0.01960
MM			
Residual cut: Single-shot	-0.0270	[-0.0867, +0.0328]	1.00000
Residual cut: PlanGate	-0.1041	[-0.1604, -0.0484]	0.03360
G56			
Residual cut: Single-shot	-0.0403	[-0.1015, +0.0200]	1.00000
Residual cut: PlanGate	-0.2975	[-0.3614, -0.2321]	0.01960
G55			
Residual cut: Single-shot	+0.0034	[-0.0472, +0.0533]	1.00000
Residual cut: PlanGate	-0.1912	[-0.2597, -0.1248]	0.01960

E Unified Seven-Model Full-Corpus Comparison

E.1 Shared population and implementation scope

The comparison includes seven models, the same 306 case identifiers, and both Single-shot and PlanGate arms, giving 4,284 final case–arm observations. All three structural metrics are available for every pair. The models share visible schemas, the eight-view output contract, and the frozen scorer.

Table S9: Recorded request settings for every full-corpus model. Output caps apply per request, not per case; Responses budgets include reasoning. All runs retain a 400-character FK plan prefix. These are heterogeneous service settings.

Model	Transport	Output cap	Temp.	Reasoning
Q30B	Chat	12,000	0.2	not sent
Q8B	Chat	24,000	0.2	low
Q32B	Chat	24,000	0.2	low
DS	Chat	24,000	0.2	not sent
MM	Chat	24,000	0.2	low
G56	Responses	16,000	not sent	low
G55	Responses	16,000	not sent	low

Service implementations differ. The GPT runs use a Responses-compatible gateway, low reasoning effort and a 16,000-token output budget including reasoning; temperature and a deterministic seed were not sent. The other five serving configurations use their recorded Chat-compatible transports and budgets. Internal JSON-format retries and the GPT selection path have not been separately reconstructed as equal-compute controls. Final-output means therefore describe the realized service/protocol combinations, while within-model differences quantify the paired changes on shared cases. Service labels are not independently authenticated immutable model snapshots.

Table S10: Final full-corpus comparison: seven models, 306 paired cases each. SS/PG means, paired differences, unadjusted 95% case-bootstrap CIs and two-sided sign-flip p use 10,000 draws. Service implementation and budget settings are reported in Table S9.

Metric	SS	PG	Δ [95% CI]	p
Q30B				
FKTopo	0.7993	0.8135	+0.0142 [-0.0178,+0.0465]	0.4025
StructIso	0.6581	0.6673	+0.0092 [-0.0146,+0.0329]	0.4487
ColTypeF1	0.8686	0.8716	+0.0030 [-0.0198,+0.0257]	0.8029
Q8B				
FKTopo	0.8129	0.8323	+0.0194 [-0.0176,+0.0563]	0.3055
StructIso	0.6416	0.6997	+0.0581 [+0.0278,+0.0899]	0.0005
ColTypeF1	0.8233	0.8728	+0.0495 [+0.0157,+0.0835]	0.0035
Q32B				
FKTopo	0.7885	0.8219	+0.0334 [-0.0050,+0.0731]	0.0943
StructIso	0.6480	0.6760	+0.0280 [+0.0025,+0.0540]	0.0317
ColTypeF1	0.8526	0.8568	+0.0042 [-0.0224,+0.0298]	0.7550
DS				
FKTopo	0.7863	0.8590	+0.0726 [+0.0333,+0.1138]	0.0006
StructIso	0.6163	0.6906	+0.0743 [+0.0387,+0.1110]	0.0002
ColTypeF1	0.8041	0.8846	+0.0805 [+0.0442,+0.1185]	0.0001
MM				
FKTopo	0.7079	0.8179	+0.1100 [+0.0649,+0.1572]	0.0001
StructIso	0.5562	0.6670	+0.1109 [+0.0726,+0.1503]	0.0001
ColTypeF1	0.8062	0.8614	+0.0551 [+0.0273,+0.0827]	0.0001
G56				
FKTopo	0.5946	0.7554	+0.1609 [+0.1049,+0.2158]	0.0001
StructIso	0.5348	0.6801	+0.1453 [+0.1080,+0.1822]	0.0001
ColTypeF1	0.8055	0.8203	+0.0148 [-0.0208,+0.0505]	0.4053
G55				
FKTopo	0.5466	0.7876	+0.2410 [+0.1845,+0.2973]	0.0001
StructIso	0.4324	0.6718	+0.2394 [+0.1935,+0.2866]	0.0001
ColTypeF1	0.8146	0.8393	+0.0246 [-0.0069,+0.0561]	0.1348

Table S11: Operational endpoints for the registered seven-model comparison. Each row uses the same 306 cases as Table S10; entries are Single-shot rate, PlanGate rate, and the paired rate difference.

Model	SQLParse			DDLExec		
	SS	PG	Δ	SS	PG	Δ
Q30B	.876	.882	+.006	.794	.807	+.013
Q8B	.814	.912	+.098	.768	.869	+.101
Q32B	.873	**.928**	+.056	.830	**.892**	+.062
DS	.817	.899	+.082	.768	.840	+.072
MM	.853	.869	+.016	.788	.804	+.016
G56	.948	.873	-.075	.824	.817	-.006
G55	.915	.863	-.052	.732	.843	+.111

Definitions and detailed case statistics. Figure S2 summarizes dispersion and joint outcomes. Tables S12–S14 provide the complete intervals and counts. The bootstrap jointly resamples 306 paired cases across the seven fixed services; its intervals quantify case-sampling uncertainty, not uncertainty over a population of models. PlanGate weakly dominates when all three paired changes are nonnegative and at least one is positive; Single-shot weakly dominates under the reverse condition. Mixed cases have positive and negative changes; exact ties have three zero changes. Strict three-metric improvement is reported separately.

Table S12: Cross-model dispersion of full-corpus mean scores. SD and range are computed across the seven fixed model services; differences are PlanGate minus Single-shot with 95% paired case-bootstrap intervals.

Metric	SD SS→PG	ΔSD [95% CI]	Range SS→PG	ΔRange [95% CI]
FKTopo	0.1000→0.0306	-0.0694 [-0.089,-0.048]	0.2663→0.1035	-0.1628 [-0.233,-0.101]
StructIso	0.0757→0.0114	-0.0643 [-0.079,-0.048]	0.2256→0.0327	-0.1930 [-0.238,-0.137]
ColTypeF1	0.0237→0.0203	-0.0034 [-0.015,0.008]	0.0645→0.0643	-0.0002 [-0.047,0.028]

Table S13: Per-metric case-level distribution of PlanGate minus Single-shot. W/T/L are the numbers of cases with positive, exact-zero, and negative paired differences; s_Δ is the case-level standard deviation.

Metric	Model	$\bar{\Delta}$	W	T	L	s_Δ
FKTopo	Q30B	+0.014	25	261	20	0.293
FKTopo	Q8B	+0.019	32	244	30	0.331
FKTopo	Q32B	+0.033	39	241	26	0.346
FKTopo	DS	+0.073	42	247	17	0.363
FKTopo	MM	+0.110	60	225	21	0.403
FKTopo	G56	+0.161	98	179	29	0.498
FKTopo	G55	+0.241	112	175	19	0.505
StructIso	Q30B	+0.009	29	254	23	0.215
StructIso	Q8B	+0.058	40	241	25	0.279
StructIso	Q32B	+0.028	40	240	26	0.230
StructIso	DS	+0.074	43	244	19	0.327
StructIso	MM	+0.111	65	220	21	0.341
StructIso	G56	+0.145	90	201	15	0.328
StructIso	G55	+0.239	111	187	8	0.417
ColTypeF1	Q30B	+0.003	35	251	20	0.203
ColTypeF1	Q8B	+0.049	37	244	25	0.302
ColTypeF1	Q32B	+0.004	34	239	33	0.235
ColTypeF1	DS	+0.081	54	236	16	0.331
ColTypeF1	MM	+0.055	77	211	18	0.249
ColTypeF1	G56	+0.015	90	169	47	0.317
ColTypeF1	G55	+0.025	117	150	39	0.284

Table S14: Mean structural-effect vectors and paired case-level joint outcomes. PG/SS dominance is weak Pareto dominance across all three metrics; parentheses within PG give the number strictly improved on all three. Mixed denotes at least one endpoint up and one down; ties are exact.

Model	ΔFKTopo	ΔStructIso	ΔColTypeF1	PG dom. (all 3+)	Mixed	SS dom. / tie
Q30B	+0.014	+0.009	+0.003	42 (16)	2	28 / 234
Q8B	+0.019	+0.058	+0.049	54 (13)	7	45 / 200
Q32B	+0.033	+0.028	+0.004	53 (17)	3	45 / 205
DS	+0.073	+0.074	+0.081	62 (32)	3	28 / 213
MM	+0.110	+0.111	+0.055	86 (45)	8	28 / 184
G56	+0.161	+0.145	+0.015	108 (63)	14	46 / 138
G55	+0.241	+0.239	+0.025	132 (88)	7	39 / 128

Three-metric Pareto comparison. All seven observed PlanGate mean vectors have positive coordinates relative to their paired Single-shot means, so each model's mean point strictly dominates its own baseline mean on the three structural endpoints. This is a descriptive mean-level relation: the model-stratified intervals in Table S10 do not exclude zero for every coordinate and do not form a simultaneous dominance test. Across the seven model choices, the non-dominated mean frontier changes from Qwen3-30B-A3B and Qwen3-8B under Single-shot to Qwen3-8B and DeepSeek-V4-Flash under PLANGATE. Figure S4 shows all paired mean vectors through their three pairwise projections. This frontier is limited to structural quality and does not include execution or package completeness.

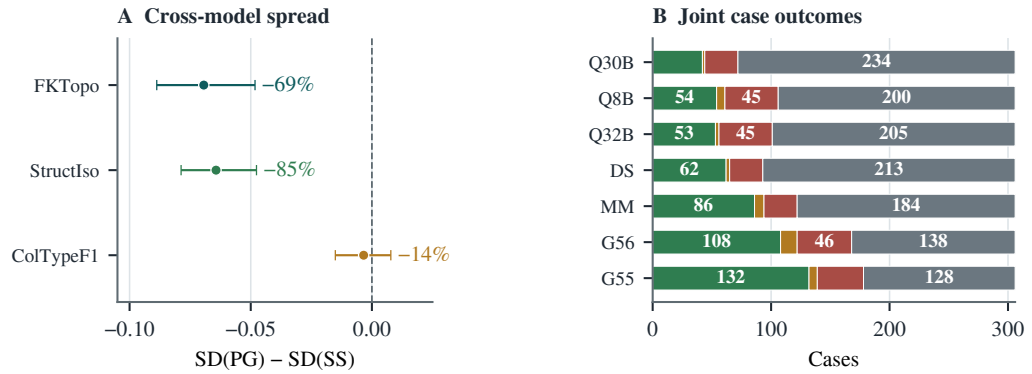

Figure S2: Case-level diagnostics for the seven registered models on the shared 306 cases. (A) Change in the standard deviation of model means with paired case-bootstrap intervals. (B) Joint three-metric case outcomes (green = PlanGate dominates, ochre = mixed, red = Single-shot dominates, gray = exact tie). Intervals are 95% intervals over paired cases; percentage labels show relative SD reductions. The two panels distinguish model-mean convergence from improvement on individual cases.

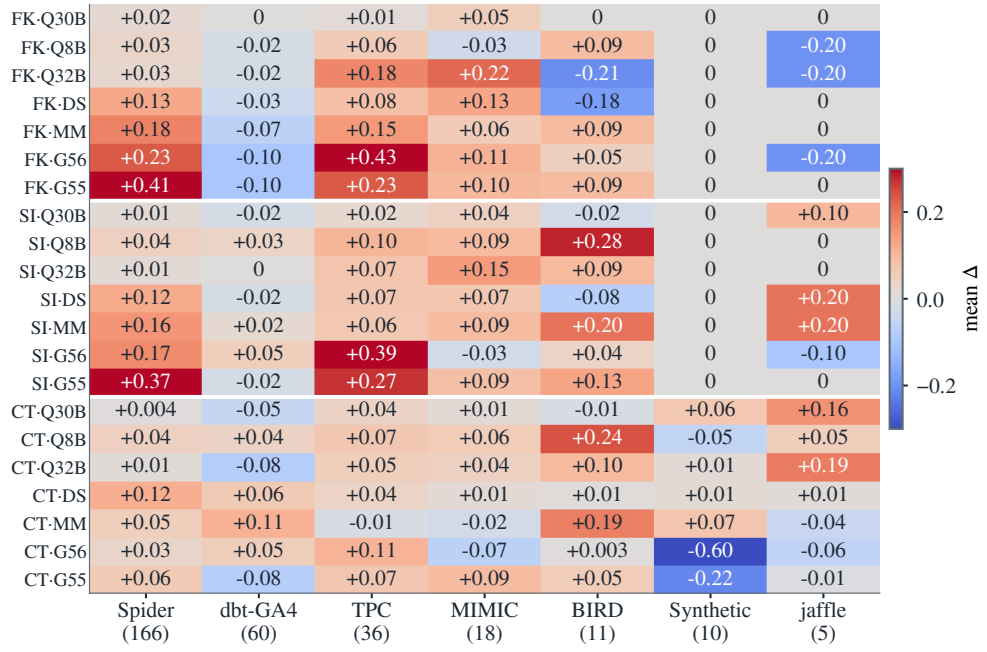

Figure S3: Source-package sensitivity of paired PlanGate–Single-shot changes. FK, SI, and CT denote FKTopo, StructIso, and ColTypeF1; columns show source packages and case counts. Labels round nonzero changes to two decimals, except values below 0.005 in magnitude, which use three; exact zeros appear as 0. Shading is clipped to $[-0.30, +0.30]$, while labels retain the full range. This is a composition breakdown, not a new-source generalization test.

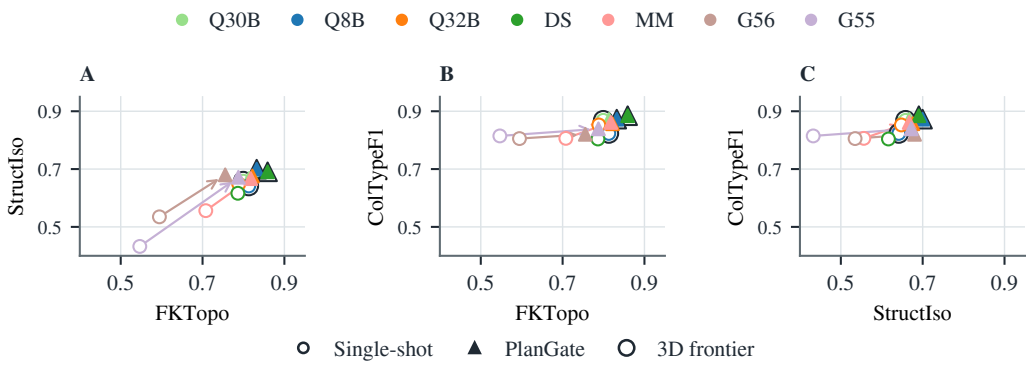

Figure S4: Three-metric Pareto comparison of seven-model means over the shared 306 cases. Each arrow connects a model's Single-shot mean (open circle) to its PLANGATE mean (triangle); color identifies the model. Axes use the same score range in all panels. Black outlines mark points on the three-dimensional non-dominated frontier within that protocol. These are observed means, not simultaneous inferential claims.

The intervention and component controls estimate FKTopo effects under their registered structural comparisons. Operational endpoints are reported separately in the main comparison and Table S11.

E.2 SEVEN-MODEL SOURCE COMPOSITION

Figure S5 relates structural recovery to evidence availability and execution. The seven-model stratum panel shows why the no-declaration subset cannot determine a universal ranking; the four-cell masked panel shows that the adapted protocol gap changes with the visible information; the execution panel separates parsing from successful SQLite execution.

The source-weighting sensitivity uses all seven fully labeled source packages (166 Spider, 60 dbt-GA4, 36 TPC, 18 MIMIC, 11 BIRD, 10 synthetic, and 5 jaffle-shop cases). Under equal package weighting, DeepSeek-V4-Flash remains the FKTopo leader and is also the StructIso leader, replacing Qwen3-8B from the case-weighted ranking. For ColTypeF1, the case-weighted leader DeepSeek changes to Qwen3-30B-A3B. DeepSeek remains the top model in all seven leave-one-package- out FKTopo and StructIso rankings; ColTypeF1's leave-one-out leaders split across DeepSeek (3/7), Qwen3-30B-A3B (3/7), and Qwen3-32B (1/7). Table S15 reports every model rank, and Figure S3 shows the corresponding per-package paired changes. Equal package weighting gives the five-case jaffle package the same weight as the 166-case Spider package; these checks therefore measure sensitivity to package composition, not expected performance on a new source.

Final-output scoring policy. Results describe the retained final outputs under the fixed scoring policy. One Qwen3-30B-A3B Single-shot output has an empty resolved graph and receives FK-Topo=1 under the scorer, while the same output does not meet the execution endpoints. Thus the structural score and execution status remain separate observations.

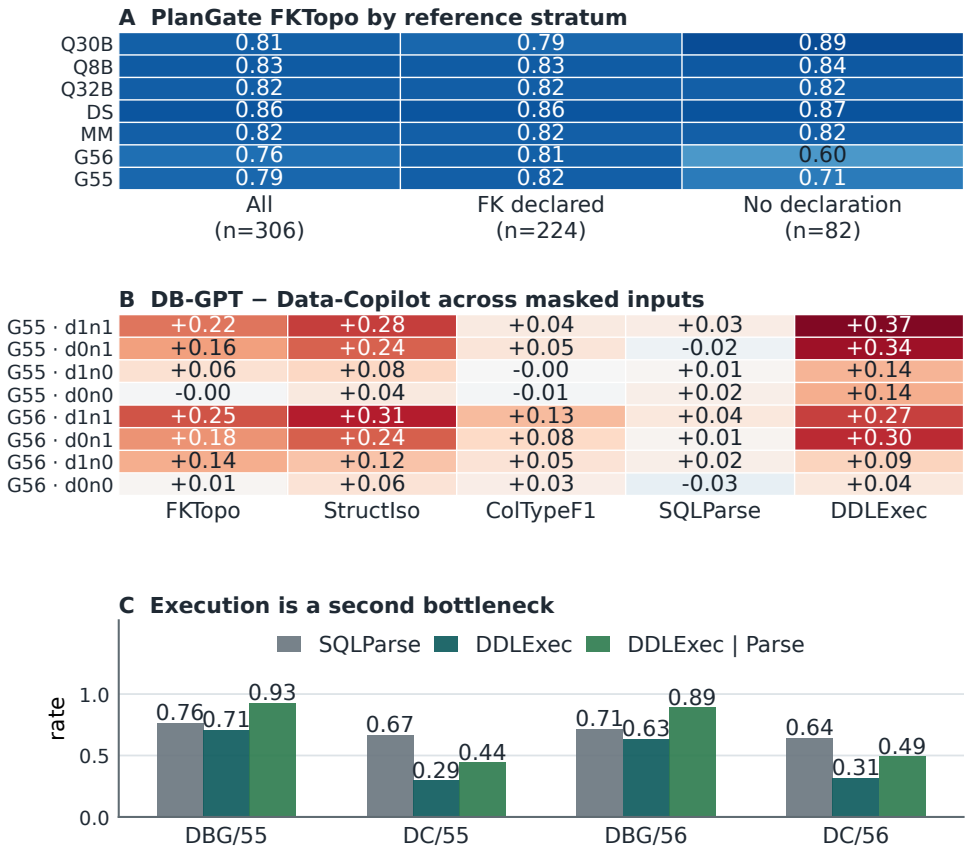

Figure S5: Structural recovery and execution. (A) PlanGate FKTopo by all, foreign-key-declared, and no-declaration strata across seven models. (B) DB-GPT minus Data-Copilot differences in the four masked information conditions. (C) SQL parsing, execution, and execution conditional on parsing. In B, d denotes declaration visibility and n meaningful column-name visibility; 1 means visible and 0 masked. Masked cells share the same case population.

Table S15: PlanGate model ranks under case weighting, equal source-family weighting, and leave-one-source-family-out sensitivity. Source-balanced ranks average the seven source-specific means equally; each leave-one-out estimate averages the remaining six families equally.

Metric	Model	Case-weighted rank	Equal-source rank	Leave-one-source rank range
FKTopo	Q30B	5	4	3–5
FKTopo	Q8B	2	5	3–6
FKTopo	Q32B	3	3	2–4
FKTopo	DS	1	1	1–1
FKTopo	MM	4	2	2–4
FKTopo	G56	7	6	4–7
FKTopo	G55	6	7	6–7
StructIso	Q30B	6	7	5–7
StructIso	Q8B	1	4	2–7
StructIso	Q32B	4	6	4–7
StructIso	DS	2	1	1–1
StructIso	MM	7	3	2–6
StructIso	G56	3	2	2–3
StructIso	G55	5	5	4–6
ColTypeF1	Q30B	3	1	1–3
ColTypeF1	Q8B	2	4	2–4
ColTypeF1	Q32B	5	3	1–4
ColTypeF1	DS	1	2	1–4
ColTypeF1	MM	4	5	5–5
ColTypeF1	G56	7	7	7–7
ColTypeF1	G55	6	6	6–6

F Baseline Comparability and Component Attribution

F.1 Expanded seven-model component controls

The component experiment separates changes in generation from changes in selection. Planned versus unplanned first candidates measure the planning contribution; replaying the same gate within each stream measures candidate selection. The difference between the gate effects in the planned and unplanned streams estimates the plan–gate interaction. Table S16 reports these matched-stream FKTopo contrasts. Only the two GPT planning contrasts remain significant after correction across 196 hypotheses. The other complete planning, gating, and interaction contrasts do not reach that threshold. This locates the clearest GPT response in planned candidate generation rather than establishing a universal benefit from gating.

Table S16: Planning and gating on matched candidate streams, final-output comparison. Entries are paired FKTopo differences. Plan: planned minus unplanned first candidate; gate: gated minus first unplanned candidate; gate after plan: gated minus first planned candidate; interaction: difference between gate effects. *Holm $p < .05$ over 196 comparisons.

Model	Plan	Gate	Gate after plan	Interaction
Q30B	+0.016	+0.007	+0.002	-0.005
Q8B	-0.006	+0.003	+0.010	+0.007
Q32B	+0.020	+0.003	+0.000	-0.003
DS	+0.028	+0.030	+0.012	-0.018
MM	+0.059	-0.003	+0.000	+0.003
G56	+0.274*	-0.029	-0.020	+0.010
G55	+0.296*	-0.026	-0.016	+0.010

The selector control compares an LLM judge with deterministic parser/FK selection on identical candidate pools, separately with and without a plan. Table S17 reports descriptive FKTopo contrasts. In the four models with complete comparisons for both pools, the direction depends on the model and pool.

Table S17: Same-pool final-output selection: LLM judge minus deterministic parser/FK selection. Each selector receives the same candidate pool. The table lists only models with complete paired coverage in both pools; ΔF is a descriptive mean FKTopo difference, not a significance test.

Model	Without a plan ΔF	With an FK plan ΔF
Q8B	-0.034	-0.011
Q32B	+0.006	+0.009
DS	-0.077	-0.021
MM	-0.028	+0.010

Additional controls vary rule-based and perturbed plans; consumed plan length (400 characters, 1,200 characters, or complete text); plan content (FKs alone versus tables, primary keys, and FKs); plan reuse; and cross-artifact coordination. Replanning shares the initial plan and first candidate, then replans from source for later candidates without candidate feedback. The coordination comparison generates eight plans, then either shares one randomly chosen plan or assigns different plans to the eight generators. These controls separate information content, context length, and shared conditioning.

All complete paired comparisons on strict package pass have zero differences, 95% intervals $[0, 0]$, and adjusted $p = 1$. This endpoint measures jointly verifiable delivery and does not distinguish the evaluated settings. Structural effects are measured separately above. Inference uses complete paired comparisons; exact support conditions are identified in the analysis records. The statistical procedure is in Appendix G.

G Statistical Analysis Plan and Decision Rules

Definitions of the additional controls. Joint masking removes explicit foreign-key declarations and replaces column names with neutral identifiers. Residual masking additionally anonymizes table and constraint names and removes comments; types and the remaining schema structure are

retained. A rule plan deterministically lists the foreign-key tuples extracted from the visible source in a fixed order, without an LLM planning call. A perturbed plan changes one recognized relation target in the consumed plan to a different existing table/column tuple of the same arity, keeping the other recognized relations unchanged. This tests whether changed plan correspondence propagates to the generated artifacts. Locked-plan conditions reuse one plan across three or five candidate slots. The refreshed-plan condition shares the initial plan and candidate but generates new source-conditioned plans for the second and third candidates, without candidate feedback. Three refreshed candidates match the three-candidate locked condition in candidate count and the five-candidate locked condition in nominal calls (six). "First" returns the first candidate; "gated" applies the same admission rule to up to three candidates.

The case is the primary statistical unit. Paired differences use case bootstrap resamples and two-sided sign-flip checks with each experiment's recorded settings. The current seven-model reanalysis uses 10,000 draws for each procedure and seed 20260915; its tests and intervals are unadjusted. Source-package cluster sensitivities remain specific to their original output sets. Exact-content clusters receive equal weight in the historical duplicate sensitivity; a 296-cluster sensitivity for the current seven-model comparison has not been computed. Accepted-subset scores are post-treatment descriptives and are not compared causally with the full deployment population.

For superiority, a two-sided interval and preregistered test must support the direction. An ordinary interval crossing zero indicates that a difference was not detected. The seven-model full-corpus rows are reported as unadjusted model-stratified estimates; the expanded controls use the correction below.

The expanded campaign uses a separate registered family of 196 comparisons, including factorial interventions, declaration-removal dose slopes, component contrasts, plan budgets, and selectors. Its case bootstrap and sign-flip tests each use 10,000 draws with seed 20260916. Each analysis retains the full 196-member Holm correction family, including unavailable comparisons; incomplete comparisons have descriptive estimates but no confidence intervals or tests. The final-output analysis evaluates retained designs rather than fixed-budget acquisition performance. The family, endpoint, and case support conditions are not changed in response to significance.