MEMORIZED OR MEMO-RIZED: PRIOR LEAKAGE OF PROCEDURAL KNOWLEDGE IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic AI systems combine foundation models' internal knowledge with external knowledge from context, retrieval, tools or memory modules. Conventionally, these are entangled with reasoning ability, making it hard to evaluate if a model truly generalizes under internal–external knowledge conflict. In this paper, we formalize knowledge into *abstract* and *surface-form* knowledge, which are components used in *reasoning*. Under this formalization, we analyze the effects of internal–external knowledge conflict and identify the **prior leakage** phenomenon: a model's *undesirable over-reliance* on internal knowledge under internal–external conflict. We evaluate 17 open source models on two newly proposed procedural knowledge datasets, SQLSHIFT and PANDASSHIFT, where common SQL and Pandas operators are swapped or renamed. We observe substantial prior leakage across the models: the knowledge utilization stays heavily anchored on internal knowledge, even when explicitly instructed to use external knowledge. We also investigated the use of gradient ascent unlearning as a way to resolve prior leakage, which was mostly unsuccessful and did not substantially change knowledge utilization. We also attempted to control internal knowledge by post-training a vintage LLM, where training with question-answer data is observed to be vital in knowledge injection while impact of training epochs and size of training set was inconclusive. We believe this evaluation suite will support the development of trustworthy AI agents that respect user-provided external evidence over unwarranted reliance on parametric priors, paving the way toward user-sovereign AI. Code is provided in supplementary material.

1 INTRODUCTION

Large Language Model (LLM)-based agents typically operate in systems that supplement the model with tools and knowledge (OpenAI, 2022). Knowledge, in this case, comes from both internal and external sources making it hard to determine the root cause when failure arises. Previous work identified three failure modes for such knowledge–reasoning systems: i) knowledge insufficiency, ii) knowledge conflict, and iii) reasoning failure, where the model is not competent enough to make the relevant connections. **Knowledge conflict** mainly concerns scenarios where external knowledge contradicts the model's internal prior. Under knowledge conflict, a model may abstain, hallucinate, return an outdated answer drawn from its internal knowledge, or an updated answer supplied externally. Knowledge conflict is especially interesting due to its temporal nature, where newer information (e.g. current affairs, new coding libraries) supersedes outdated versions. Context-dependent associations can also produce contradictory outcomes: in oncology, rising white blood cell count after chemotherapy can indicate bone marrow recovery from treatment-induced neutropenia, but in other contexts can indicate infection, leukemia or sepsis (Cleveland Clinic, 2022).

In LLM literature, knowledge conflict has primarily been studied in the context of *factual* knowledge, where entities in question-answering pairs are replaced with their counterfactual versions to induce conflicts (Qian et al., 2024; Xie et al., 2024; Cheng et al., 2024; Jin et al., 2024). Under these settings, LLMs are usually evaluated based on their final task accuracies, memorization ratios (Longpre et al., 2021; Chen et al., 2022; Xie et al., 2024; Jin et al., 2024) or knowledge-preference scores (Kim et al., 2026). In contrast, research on how LLM-based agents handle procedural knowledge has so far focused on out-of-domain (OOD) settings, namely whether a model can *extend* its skills to unfamiliar

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.

problems, such as esoteric programming languages (Sharma & Chopra, 2026), hypothetical scientific laws (Zheng et al., 2026), unseen rule-following tasks (Ma et al., 2025), or modified real-world tasks including code generation (Wu et al., 2024). Less attention has been paid to conflicts in ***procedural*** knowledge, where the basic knowledge elements underlying a procedure are replaced with counterfactual alternatives. Focusing on procedural knowledge, we identify three research gaps:

(G1) Semantic equivalent conflicts. Existing datasets construct conflicts by substituting semantically inequivalent knowledge elements (e.g. "Sorting a list of lists based on first element" for "Sorting a list of lists based on second element"); there lacks attention on whether models can resolve conflicts between different surface-form representations of the same underlying semantic concept.
(G2) Latent conflict dynamics. Existing metrics based on absolute deviations in final answers, such as memorization ratio and knowledge preference, disregard shifts in the model's internal probability mass and therefore may fail to capture subtle interactions between internal and external knowledge.
(G3) Relationship between training exposure and knowledge utilization undetermined. Existing studies rely on modern LLMs with opaque training recipes that might already possess relevant knowledge. As such, the amount of internal knowledge is neither known nor controllable, and how training exposure shapes a model's "inertia" toward external knowledge remains under-studied.

This paper addresses these gaps by making the following contributions. First, we formalize procedural knowledge and identify the phenomena of ***prior leakage***, where LLMs overly rely on internal knowledge during a conflict. Second, we further introduce two new metrics to evaluate knowledge conflict: *knowledge anchor* and *knowledge shift*, which quantifies a model's utilization of internal and external knowledge in its probability mass, and propose two new datasets, SQLSHIFT and PANDASSHIFT for evaluating the leakage of procedural knowledge via controlled counterfactual experiments. Third, across an extensive study over 17 open source models, our experiments reveal that prior leakage is *pervasive* across all models, and simple unlearning of internal knowledge (prior) does not lead to an expected decrease in prior leakage, highlighting the need for future research into methods for mitigating procedural prior leakage in LLMs.

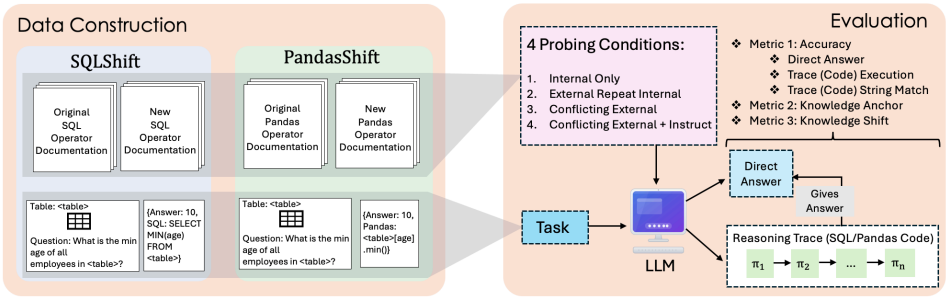

Figure 1: Overview: Each model is provided with: a task (includes a database table and question which requires the model to return an aggregated value from the table and its associated code) and documentations of either the original or new versions of SQL/Pandas Operators, depending on the probing condition (see Figure 3 for example). We evaluate both output answer and code.

2 RELATED WORK

Knowledge in Large Language Models. Knowledge in LLM-based agents can be categorized into **internal (parametric)** knowledge, held in the weights, and **external** usually non-parametric knowledge, supplied at inference (Zhang et al., 2026). Internal knowledge is acquired through large-scale pre-training on massive web corpus (Liu et al., 2025; Geva et al., 2021b) and later reshaped by post-training techniques, such as continued pre-training (Gupta et al., 2023), supervised fine-tuning (Lu et al., 2025), and parameter-efficient fine-tuning (Hu et al., 2022; Biderman et al., 2024), by knowledge editing (Meng et al., 2022), or removed by machine unlearning (Jang et al., 2023; Yao et al., 2024; Zhang et al., 2024). External knowledge is provided in-context (Dong et al., 2024; Brown et al., 2020), through retrieval-augmented generation (Lewis et al., 2020) with lexical (Robertson & Zaragoza, 2009), semantic, or hybrid retrieval (Cormack et al., 2009), tool calling (Qin et al., 2024), or dedicated memory models like MeMo (Quek et al., 2026).

Internal vs. External Knowledge. There is no definitive conclusion on which type of knowledge LLMs favor under conflict: Xie et al. (2024) find that, with simple entity substitution, models tend

to pick the original version from internal knowledge even when instructed otherwise, but coherent counterfactual evidence yields answers mostly consistent with external knowledge; Qian et al. (2024) and Cheng et al. (2024) report opposing observations. LLMs are sensitive to how external evidence is presented, display herding toward the side with more supporting evidence and are sensitive to evidence ordering (Xie et al., 2024; Jin et al., 2024). As knowledge popularity increases, internal knowledge utilization under conflict increases, up to as much as 80% (Xie et al., 2024; Cheng et al., 2024; Jin et al., 2024). Common metrics include memorization ratio $f_i/(f_e + f_i)$ (Xie et al., 2024; Jin et al., 2024; Longpre et al., 2021), knowledge preference (Kim et al., 2026), and output confidence (Jin et al., 2024; Qian et al., 2024). Conflict datasets are typically question-answer (QA) datasets with counterfactual versions (ConflictQA (Xie et al., 2024), EchoQA (Cheng et al., 2024)); Wu et al. (2024) introduce 11 counterfactual versions of real-world tasks including code generation. Other work probes whether LLMs can generalize knowledge to unfamiliar settings: EsoLang-Bench (Sharma & Chopra, 2026), NewtonBench (Zheng et al., 2026), and KOR-Bench (Ma et al., 2025). Kim et al. (2026) study how training conditions shape context reliance, but on synthetic data whose underlying knowledge (entity and relation recognition) is not new to the model. Full analysis is in Appendix H.

3 Formalizing Prior Leakage

Bloom's Taxonomy of Learning classifies knowledge into four types (Stapleton-Corcoran, 2023): factual, **procedural**, conceptual, and metacognitive, where procedural knowledge refers to the methods, techniques, and algorithms. Our work on knowledge conflict focus on procedural knowledge acquired by LLMs. We consider knowledge from a two-layered perspective, visualised in Figure 2.

Definition 1 (Abstract Knowledge) We define abstract knowledge as $\mathcal{K}_\mathcal{A} = \{k^1_\mathcal{A}, \ldots, k^n_\mathcal{A}\}$ where each element $k^i_\mathcal{A}$ is an atomic element and sits at the most fundamental level of the concept hierarchy (e.g. addition, subtraction in Mathematics). Abstract knowledge should remain relatively constant, but their quantity could increase slowly as our overall understanding of the world expands e.g. discovery of new chemical elements, new animal species identified.

Definition 2 (Surface-Form Knowledge) We define surface-form knowledge as concrete instantiations of abstract knowledge $\mathcal{K}_\mathcal{A}$ (e.g. addition can be represented by the English word "plus" or by the symbol $+$). Given a surface-form knowledge $\mathcal{K}_\mathcal{S}$, we denote its elements as $\{k^1_\mathcal{S}, \ldots, k^n_\mathcal{S}\}$, and assume there exist operations mapping the surface form to the abstract form and vice versa. Notably, a single abstract knowledge atom can be instantiated into multiple surface forms: the abstract notion of the procedure "taking an average" may be represented as `AVG()` in SQL or `.mean()` in Pandas. In the context of LLMs, the surface form knowledge can come from internal knowledge, acquired during the model's training, or from external knowledge supplied in-context or by a retrieval system.

Our knowledge elements are related to the Knowledge Ingredients (KIs) (Li et al., 2026a), as both represent atomic units of knowledge. However, KIs are natural-language statements extracted from reasoning traces, whereas we distinguish abstract knowledge from its surface-form realizations.

Definition 3 (Reasoning) Given the surface-form knowledge $\mathcal{K}_\mathcal{I}$ acquired from pre-training, $\mathcal{K}_\mathcal{E}$ supplied externally, and task $\mathcal{T}$, reasoning is a set of transformations that map elements from $\{\mathcal{K}_\mathcal{I}, \mathcal{K}_\mathcal{E}\}$ to the corresponding atoms in the abstract knowledge repository $\mathcal{K}_\mathcal{A}$ and ordering them correctly, such that when mapped back to the surface form, it returns a reasoning trace that leads to the correct answer: $\mathcal{R}(\mathcal{K}_\mathcal{I}, \mathcal{K}_\mathcal{E}, \mathcal{T}) = \pi,$ where π is the reasoning trace,$\forall \pi_i \in \pi,\ \pi_i \in \mathcal{K}_\mathcal{I} \vee \pi_i \in \mathcal{K}_\mathcal{E}$

We refer to a model's internal knowledge as its **prior**, inspired by the concept of prior in Bayesian modeling (Murphy, 2023; McElreath, 2020). The tendency to rely on prior knowledge, which we call *prior inertia*, can be beneficial or harmful depending on the context: when externally provided information is untrustworthy, relying on the prior may be preferable. However, when external knowledge provides more up-to-date information, it is desirable not to rely on the prior. Here, we focus on cases where prior inertia is undesirable, which we refer to as **prior leakage**.

Definition 4 (Prior Leakage) Given a task $\mathcal{T}$, the model is provided surface-form knowledge $\mathcal{K}_\mathcal{E}$ in the context, i.e., external knowledge. The model has also acquired a surface-form knowledge $\mathcal{K}_\mathcal{I}, \mathcal{K}_\mathcal{E} \neq \mathcal{K}_\mathcal{I}$ from (pre-)training, i.e., internal knowledge. Given that $\mathcal{K}_\mathcal{E}$ and $\mathcal{K}_\mathcal{I}$ refer to the same underlying abstract knowledge, if the model over-relies on $\mathcal{K}_\mathcal{I}$ to complete the task $\mathcal{T}$ rather than using the provided $\mathcal{K}_\mathcal{E}$, we say **prior leakage** has occurred, i.e. when the model is provided with $\mathcal{K}_\mathcal{I}$ implicitly and $\mathcal{K}_\mathcal{E}$ explicitly, the reasoning trace utilize $\mathcal{K}_\mathcal{I}$ instead of $\mathcal{K}_\mathcal{E}$: $\mathcal{R}(\mathcal{K}_\mathcal{I}, \mathcal{K}_\mathcal{E}, \mathcal{T}) = \pi, \quad \forall \pi_i \in \pi,\ (\pi_i \in \mathcal{K}_\mathcal{I}) \wedge (\pi_i \notin \mathcal{K}_\mathcal{E})$. Since a model's output is unlikely to be governed purely

by external knowledge, we quantify the relative influence of internal and external knowledge by comparing the probability assigned to each trace token π_t, conditioned on with and without externally supplied knowledge: $P(\pi_t \in \mathcal{K}_\mathcal{I} \mid \emptyset), P(\pi_t \in \mathcal{K}_\mathcal{E} \mid \emptyset), P(\pi_t \in \mathcal{K}_\mathcal{I} \mid \mathcal{K}_\mathcal{E}), P(\pi_t \in \mathcal{K}_\mathcal{E} \mid \mathcal{K}_\mathcal{E})$, where $\emptyset$ denotes the absence of externally supplied knowledge.

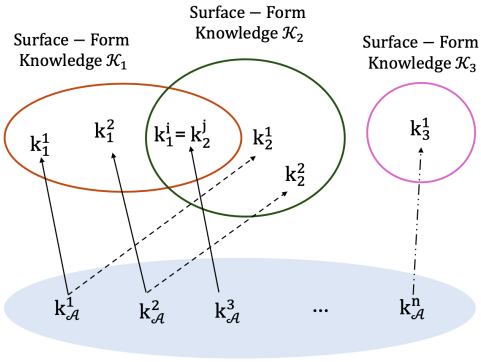

Figure 2: Abstract knowledge elements $\{k_\mathcal{A}^1, ..., k_\mathcal{A}^n\}$ and surface-form knowledge $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ and their respective elements. The surface-form knowledge may overlap, or be mutually exclusive. Each $k_\mathcal{A}^i$ can be mapped to a surface-form knowledge or not at all.

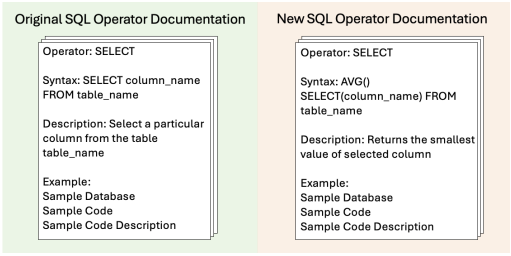

Figure 3: Examples of SQL documentation supplied both implicitly and explicitly to the model. Left gives the original version of SQL documentation for the `SELECT` operator and right gives new version: Here, the underlying $k_\mathcal{A}^i$ which represents taking the minimum, is represented by `MIN()` in the original version and `SELECT` in the new version. $k_\mathcal{A}^j$ representing selection is represented by `SELECT` in the original version and `AVG()` in the new version.

4 Prior Leakage Evaluation Framework

4.1 Dataset Construction

Most existing work on knowledge conflict has focused on open-book factual knowledge (Joshi et al., 2017; Mallen et al., 2023; Cheng et al., 2024). Our benchmark focuses on procedural knowledge in coding, which is an ideal representation of the interactions between abstract and surface-form knowledge. Each coding operator describes a single abstract knowledge concept, and each operator itself is a surface-form knowledge element, for a particular coding language. Following SciTrek (Li et al., 2026b), which generates templated SQL queries of varying difficulty and converts them to natural language (enabling ground-truth verification via string matching or code execution), we extend this approach to Pandas, constructing two new datasets: SQLShift and PandasShift. Each contains (i) natural language QA pairs with ground-truth SQL or Pandas queries, and (ii) tutorial documentation defining the operators used to compose them. Figure 1 gives an overview.

For the query part, we adopt the simplest SciTrek template (Li et al., 2026b). The ground-truth procedure is described by a single aggregation operator selected from {MIN, MAX, SUM, AVG} and applied to one integer-typed column in a database, using the template: `SELECT Operator(Column) FROM table`. Its natural language form is composed using a template "What is the {Operator} {Column} of all {Table}?" The Python Pandas variant uses `DataFrame` API instead. We randomly generate 10 tables, each with 2–3 integer columns, and instantiate the template once per aggregation operator per integer column, giving 104 evaluation questions in total. Ground-truth answers are obtained by running SQL using SQLite and Pandas using Python.

For the external knowledge part, we use explicit documentation for each operator, including its name, function description, syntax, and sample use cases. For controlled experiments, we construct two versions of the external knowledge: an **original version** that provides the operators' conventional definitions, consistent with the model's internal knowledge $\mathcal{K}_\mathcal{I}$, and a **new (hypothetical) version** that assigns each operator a counterfactual meaning, representing $\mathcal{K}_\mathcal{E}$. The original documentation uses conventional operator definitions, obtained from W3Schools (W3 Schools, 2026) for SQL and the official Pandas documentation (pandas, 2026) for Python Pandas. To construct the new (hypothetical) documentation, we modify either the operator semantics or its surface form: we reassign a familiar operator to a different meaning (e.g., redefining SQL `SELECT` to compute the minimum `MIN()`) or replace a familiar operator name with an entirely new name of identical length (e.g., replacing symbol `SELECT` with symbol `TFMFDU`). Figure 3 provides an example for SQLShift. In addition,

we vary the number of sample use cases provided per operator (1, 5 or 10) to control the amount of external evidence supporting the documented operator semantics. To manipulate internal knowledge in the vintage LLM (see Section 7), we additionally construct a training set of 688 SQL QA pairs using tables from WIKISQL (Zhong et al., 2017), following the same question-generation procedure as SQLSHIFT. Example questions, sample data are provided in Appendix A.

4.2 PROBING CONDITIONS

To determine the effects of an internal–external conflict in procedural knowledge, we measure and compare the reasoning traces of the model, which is the output code, with and without (baseline) conflict. Specifically, we probe the model under four conditions (refer to Figure 1), organized along two axes: whether external knowledge is present, and, where present, if it is consistent with or contradicts internal knowledge. *Internal Only:* the task $\mathcal{T}$ is not provided any surface-from knowledge externally; the output reflects purely $\mathcal{K}_\mathcal{I}$. *External Repeat Internal:* the internal knowledge $\mathcal{K}_\mathcal{I}$ is repeated by including the original documentation of each element in-context. *Conflicting External:* the new surface-form knowledge $\mathcal{K}_\mathcal{E}$ is supplied in-context by including the new documentation in the input. If leakage occurs here, providing external knowledge in context is insufficient to override the internal prior under ordinary prompting. *Conflicting External + Instruct:* the same new knowledge $\mathcal{K}_\mathcal{E}$ is supplied, paired with an explicit instruction directing the model to only use the external knowledge. The contrast between the latter two separates the inability to use external evidence without explicit instruction from a more fundamental tendency to weight the internal prior over external evidence more heavily. Full prompt template for each condition is provided in Appendix A.

4.3 EVALUATION METRICS

Existing knowledge-conflict studies primarily evaluate models using the memorization ratio (Xie et al., 2024; Jin et al., 2024; Longpre et al., 2021), which measures how often a model chooses internal over external knowledge. However, this metric offers little insight into the model's exact knowledge utilization. We address this gap with two new metrics, **Knowledge Anchor** and **Knowledge Shift**, to quantify the exact usage of internal and external knowledge under conflict.

Metric 1: Accuracy Since the model emits both a direct answer and a reasoning trace (the SQL query/Pandas code), we evaluate both against the ground truth. As the underlying abstract knowledge is the same, the direct answers should stay the same, while the reasoning trace should be faithful to the knowledge requirements of each condition.

Metric 2: Knowledge Anchor (KA) Providing external knowledge in-context may be insufficient to produce traces entirely aligned with $\mathcal{K}_\mathcal{E}$, but may still induce a probability distribution shift in the reasoning trace. KA thus identifies the utilization of $\mathcal{K}_\mathcal{I}$ and $\mathcal{K}_\mathcal{E}$ for each trace token π_i:

$$\text{Knowledge Anchor (KA)} = \frac{P(\pi_i \in \mathcal{K}_\mathcal{E} \mid \Omega, \pi_{1:t-1}) - P(\pi_i \in \mathcal{K}_\mathcal{I} \mid \Omega, \pi_{1:t-1})}{0.5\big[P(\pi_i \in \mathcal{K}_\mathcal{I} \mid \Omega, \pi_{1:t-1}) + P(\pi_i \in \mathcal{K}_\mathcal{E} \mid \Omega, \pi_{1:t-1})\big]}, \tag{1}$$

where Ω is the input prompt excluding the task ($\emptyset$ for *Internal Only*, $\mathcal{K}_\mathcal{I}$ for *External Repeat Internal*, $\mathcal{K}_\mathcal{E}$ for *Conflicting External*, $\{\mathcal{K}_\mathcal{E}, \mathcal{I}\}$ for *Conflicting External + Instruct*). The metric was modified from the midpoint formula for price elasticities (Girante, 2026); it captures the relative distribution of the model's knowledge without being sensitive to absolute probability values. KA is bounded in $[-2, 2]$: -2 is full internal knowledge utilization and $+2$ is full external knowledge utilization.

Metric 3: Knowledge Shift (KS) Knowledge Shift compares how the model's probability mass changes by comparing *Internal Only* against the other 3 the conditions. For *External Repeat Internal* it measures $\Big(P(\pi_i \in \mathcal{K}_\mathcal{I} \mid \mathcal{K}_\mathcal{I}) + 1\Big) / \Big(P(\pi_i \in \mathcal{K}_\mathcal{I} \mid \emptyset) + 1\Big)$, *Conflicting External* and *Conflicting External + Instruct* measures $\Big(P(\pi_i \in \mathcal{K}_\mathcal{E} \mid \Omega) + 1\Big) / \Big(P(\pi_i \in \mathcal{K}_\mathcal{E} \mid \emptyset) + 1\Big)$, Ω refers to the same as in KA. It seeks to determine how much inertia there is in the model, i.e. whether knowledge utilization shifts at all. Full derivation is in Appendix B.

5 EVALUATING PRIOR LEAKAGE IN MODERN LLMS

Setup. We evaluate 17 models across 6 model families (Llama 3.1 (Grattafiori et al., 2024), Qwen2.5 (Qwen et al., 2024), Qwen2.5-Coder (Hui et al., 2024), Qwen3 (Yang et al., 2025) and Qwen3-Coder (Cao et al., 2026)) at parameter scales from 4B to 405B, covering general-purpose

Instruct models, Coder variants fine-tuned specifically for code generation, and Thinking models. The full list of models is in Table 5. We run each model under the 4 probing conditions and the two datasets. For SQLSHIFT we run the full set of experiments, varying the number of samples per operator documentation (1/5/10) for conditions *External Repeat Internal, Conflicting External, Conflicting External + Instruct* and only run up to 5 samples per operator documentation for Llama-3.1-405B-Instruct as 10 samples exceeded machine memory. For PANDASSHIFT we tested with only 5 samples per documentation. *Conflicting External* and *Conflicting External + Instruct* were additionally ran across 6 different versions of external knowledge (five derangement swaps and one new-name version). For each question the model is required to output both a direct answer and the corresponding SQL/Pandas code (the reasoning trace). For every trace token we record the model's token probability distribution, filtered to a cumulative (top-p) threshold of 0.9999. We then locate the token position of selection and aggregation operators and obtain the probabilities for the original and new operators. Given that each question has two operators (one selection and one aggregation), this gives $104 \times 2 = 208$ measurements per condition, which we then take the average.

Finding 1: Prior leakage exists with varying levels in all modern models. Figure 4 gives the accuracy results for all 17 models across the 4 probing conditions, verified against the original and new version of operator documentation. We observe that across all models, none are able to fully utilise the external knowledge, even when explicitly instructed to do so. In the full per-family accuracy results in Appendices D and E, we additionally note that while most do not attain high performance on the full replacement string match (New Str), they display higher utilization of the new selection operator. This is because aggregation operators only appear in aggregation-operator documentation, but selection operators appear in both. Example LLM outputs are shown in Table 6.

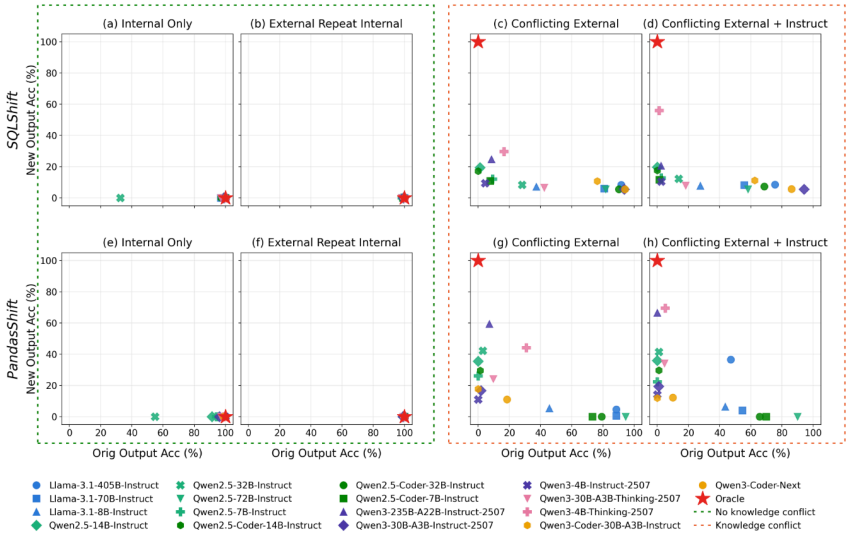

Figure 4: Performance of models from executing the SQL query/Pandas code returned. x-axis gives the accuracy using the original versions (internal knowledge) of operator, y-axis gives the accuracy when compared against the new versions (external knowledge). Oracle (★) represents the ideal outcome: if there is no prior leakage, models should have 100% accuracy for original versions output and 0% accuracy for new version under *Internal Only* and *External Repeat Internal*, and vice versa under *Conflicting External* and *Conflicting External + Instruct*. Under *Internal Only* most models are clustered at 100% accuracy for the original version and 0% accuracy for the new version, which is expected since the new version documentations are not supplied at all. In *External Repeat Internal*, the clustering is even more prominent. When the new version is supplied to the model for *Conflicting External* and *Conflicting External + Instruct* the models' performance on the new version increases and correspondingly their performance for the original version decreased, but none attain 100% accuracy for the new version, indicating prior leakage.

Finding 2: When prior leakage exists, the models never fully utilise the external knowledge. Prior leakage exists because the underlying knowledge shift is insufficient. From Figure 5, we see that Knowledge Anchor remains almost purely in internal knowledge across the highest-leakage models: Qwen3-30B under SQLSHIFT and Qwen2.5-72B under PANDASSHIFT. While the instruct

prompt increases external knowledge utilization as indicated by shifts in the knowledge anchor, it remains below 2 or even negative in some cases, indicating that the model does not fully recognize, or only slightly admits, the external knowledge. This is further confirmed by the Knowledge Shift metric: where models with a negative Knowledge Anchor, have Knowledge Shift of close 1 (i.e. the model did not utilize it much of the external knowledge). Full results are in Tables 13, 14, 21 and 22.

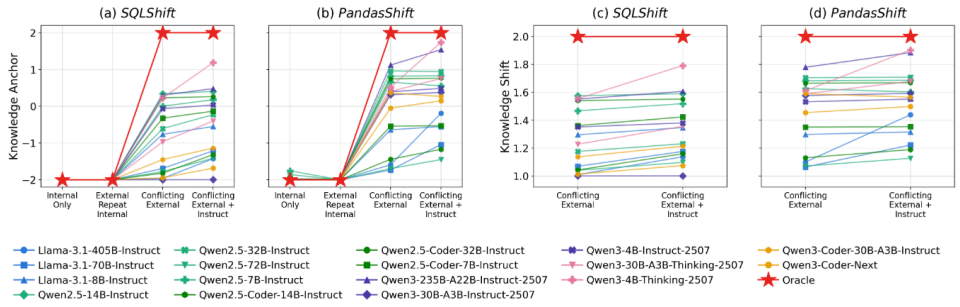

Figure 5: Knowledge Anchor and Knowledge Shift for all 17 models under the 4 probing conditions and 2 datasets. Each line represents a single model. Oracle (★) represents the ideal outcome where a model does not exhibit prior leakage. For Knowledge Anchor, the values lie between [-2, 2], -2 indicates full internal knowledge utilization and 2 represents full external knowledge utilization. An oracle (★) should display full internal knowledge utilization when no external knowledge is supplied and vice versa when external knowledge is provided. We observe none of the models fully utilize external knowledge under conflict, and some models mainly utilize internal knowledge even when instructed to use external knowledge. For Knowledge Shift, we observe that while there is an increase in external knowledge utilization from *Conflicting External* to *Conflicting External + Instruct*, the overall shift compared to *Internal Only* is not huge: most models lie below 1.4 for SQLSHIFT.

Finding 3: LLMs are more receptive to external knowledge under PANDASSHIFT than under SQLSHIFT. Comparing the two rows of Figure 4, under PANDASSHIFT the models achieve higher utilization of external knowledge, achieving higher scores for the new output accuracy. Hasegawa et al. (2026) found that LLMs consistently achieved double the accuracy for Python compared to SQL on the same tasks and that with SQL, it generated more syntax errors and incorrect column selection, aligning with our findings. On the other hand, Hu et al. (2026), using more complex templates, found better performance with SQL than Python.

Finding 4: Context-faithful prompt increases utilization of the external knowledge, but it is not perfect. Looking at Figures 4 & 5, we observe that across families, under *Conflicting External + Instruct* the models exhibit lower prior leakage than under *Conflicting External*, with varying degrees. For Qwen3-30B-A3B-Instruct-2507 the additional instruction made no difference to SQLSHIFT string-match accuracy, though its Knowledge Anchor changed from -1.9992 to -1.9975 and Knowledge Shift from 1.0003 to 1.0006. On the other hand, for Llama-3.1-70B-Instruct the string-match accuracy increased from 5.56% to 8.23%. Context-faithful prompting is not a perfect one-size-fits-all strategy and in most models the Knowledge Anchor is still highly negative.

Finding 5: Increasing model size reduces prior leakage initially, but causes it to increase after. Models exhibiting the highest prior leakage are not the smallest in their families: Llama-3.1-70B-Instruct, Qwen2.5-72B-Instruct, Qwen2.5-Coder-32B-Instruct, Qwen3-30B-A3B-Instruct-2507 and Qwen3-30B-A3B-Thinking-2507 for SQLSHIFT, and similar for PANDASSHIFT; only Qwen3-4B-Instruct-2507 is a smallest-in-family model with highest leakage. Mid-sized models (Qwen2.5-14B-Instruct, Qwen2.5-Coder-14B-Instruct, Qwen3-Coder-30B-A3B-Instruct) tend to have the least leakage, while larger models performed worse. Longpre et al. (2021) similarly noted that larger models displayed greater memorization, while Su et al. (2024) made a differing observation.

Finding 6: Model fine-tuning can affect a model's prior leakage: Thinking models displayed lower leakage, Coding models displayed relatively higher leakage. By comparing the performance of models across families, we observe that thinking models exhibit lower occurrence of prior leakage. In particular, the Qwen3-4B-Thinking-2507 model achieved a Knowledge Anchor of 1.2288 and 1.7404 under *Conflicting External + Instruct* for SQLSHIFT and PANDASSHIFT respectively. However, the coding models tend to display higher levels of leakage, when compared to other models of similar parameter size, exhibiting greater "inertia" in adopting external knowledge.

Finding 7: More examples in the operator documentation reduce prior leakage in larger models, but not necessarily in smaller models. In Figure 6, we observe that as the number of samples increases from 1 to 5 then 10, the Knowledge Anchor initially increases (leakage decreases) then decreases in some of the smaller models (4B-32B). On the other hand, the larger models (70B-405B) exhibits decreasing leakage as samples increase from 1 to 10 more consistently. Including more sample cases increases context length, which may make it harder for smaller models to elicit the main points; this is in line with the "over-prompting" phenomenon observed by Tang et al. (2025).

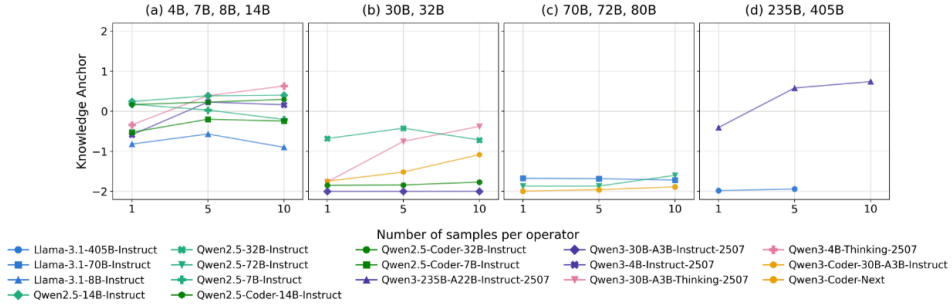

Figure 6: How the number of samples (1/5/10) in each operator documentation affects prior leakage in terms of Knowledge Anchor across model sizes under *Conflicting External* condition.

6 Can prior leakage be resolved with unlearning?

To investigate if prior leakage is due to some underlying inherent property of models or it is due to persistence of internal knowledge in models, we perform unlearning as a way to resolve prior leakage.

Table 1: Summary of the gradient-ascent unlearning experiment on SQLShift. Success of unlearning is given by the *Internal Only* string match and code execution accuracy. We evaluate prior leakage using Knowledge Anchor on the *Conflicting External* and *Conflicting External + Instruct* conditions (-2 gives full internal knowledge utilization and 2 gives full external knowledge utilization). Oracle represents ideal outcome after unlearning: model returns 0% acc. when evaluated against original versions since it no longer possess relevant knowledge and fully utilizes external knowledge provided.

	Internal Only accuracy (%)		Knowledge Anchor	
Model	**Orig. Output**	**Orig. Str**	***Conflicting External***	***Conflicting External + Instruct***
Qwen2.5-32B-Instruct	0.96	0.96	-1.6094	-1.6664
Qwen2.5-Coder-32B-Instruct	99.04	99.04	-1.9519	-1.8083
Qwen3-30B-A3B-Instruct-2507	99.04	0.00	-2.0000	-2.0000
Qwen3-30B-A3B-Thinking-2507	98.08	97.12	-1.7774	-1.0996
Qwen3-Coder-30B-A3B-Instruct	100.00	76.92	-1.8218	-1.5782
Oracle	0.00	0.00	2	2

Setup Gradient ascent unlearning was performed on the 30B/32B models for the SQLShift task following Jang et al. (2023), with a max loss of 20 for early stopping and ran for up to 10 epochs. We use the same SQL documentation in a QA setting ("What does the {operator_name} do?" with the documentation as the answer) so the model does not simply unlearn text-completion abilities.

Finding: Gradient ascent unlearning was not successful in removing all relevant SQL knowledge, models were still able to complete the task accurately under pure internal knowledge. Referring to Table 1, we observe high accuracy for most models under *Internal Only*. In fact, the models successfully unlearned the pure-upper-case SQL format but not mixed-case queries (string match is case-sensitive but SQLite is not), so we consider gradient ascent unlearning success using both execution and string-match metrics. Coder models unlearned SQL knowledge less than non-Coder models. Where unlearning was most significant (Qwen2.5-32B-Instruct), prior leakage *increased*, contrary to our expectation. We cannot identify if this was due to unsuccessful gradient ascent unlearning, because it only unlearned the exact presented format, or because the model is simply incompetent at knowledge mapping. Since the definition of successful forgetting is still not well-defined (Peleg et al., 2026; Wei et al., 2025) we thus are unable to make strong conclusions that negate the presence of prior leakage. Full results in Appendix F.

7 EVALUATING PRIOR LEAKAGE IN VINTAGE LLM

Setup How training shapes internal knowledge, and hence prior leakage, cannot be answered with modern LLMs, since we have no way of knowing or controlling how much exposure any given pre-trained model received on a specific topic. With a vintage LLM, Talkie (Levine et al., 2026), we can select a target domain (e.g. coding) where the model possesses no internal knowledge. We inject SQL knowledge into Talkie in two phases: **Phase 1** acquires the original knowledge $\mathcal{K}_{\mathcal{I}}$ through continued pre-training on SQL operator documentation; **Phase 2** performs supervised fine-tuning using LoRA on QA pairs to teach the model to apply that knowledge in the exact evaluation format. We control internal knowledge along three axes: the volume of corpus (documentation examples in Phase 1; QA pairs in Phase 2), the type of corpus (no training/only Phase 1/only Phase 2/both phases), and the number of epochs. We vary these parameters and investigate how they affect internal knowledge and consequently prior leakage, adapting code optimization from RicardoDominguez (2026). After training, we test for prior leakage using the four conditions, but use one sample per operator, since increasing this produced erroneous results. Full training details in Appendix G.1.

Finding 1: Training with QA data was more effective in SQL knowledge acquisition for Talkie, as compared to training only using SQL operator documentation. Table 2 presents the best parameters for each type of training. Pure Phase 1 training and no training were excluded as the model was simply unable to follow the output requirements, returning long-form sentences and no SQL query (0% accuracy). For pure Phase 2 the best performance was 10 epochs on 100 training pairs; for full training it was 10 epochs in Phase 1 with 1 sample per operator documentation and 5 epochs for Phase 2 on 100 training pairs. The SQL query string match has a lower accuracy than executing the query because string match is case-sensitive while SQLite is not.

Table 2: Accuracy (%) for Talkie in the *Internal Only* condition, at the best checkpoint of each regime. Phase 1 **Ex.** = sample use cases per operator in the documentation; Phase 2 **QA** = number of QA training pairs; "–" = phase not run. Full results in Appendix G.

	Phase 1		Phase 2				
Regime	**Epochs.**	**Samples per Op.**	**Epochs.**	**Num QA Train Data**	**Direct**	**Orig Str**	**Orig Out**
Phase 2 only	–	–	10	100	12.50	73.08	89.42
Full	10	1	5	50	9.62	71.15	81.73

Finding 2: Effects of number of epochs training-set size is mixed. Across all trained Talkie checkpoints, none utilized the new operators, resulting in 0% accuracy for New Str and New Output. In fact, we do not observe any consistent trends in relation to the number of epochs and size of training-set/number of samples per operator documentation. Full results in Appendix G.3, G.4.

8 CONCLUSION & DISCUSSION

Summary. In this paper, we study prior leakage in procedural knowledge, which is the over-reliance on internal knowledge under procedural knowledge knowledge conflict in LLM-based agents. We introduce two new datasets, SQLSHIFT and PANDASSHIFT, and new metrics, Knowledge Anchor and Knowledge Shift, to evaluate prior leakage. On both datasets, we find that prior leakage is significant across 6 LLM model families, and knowledge stays anchored on the internal prior, never completely utilizing external knowledge even when explicitly instructed to do so. Gradient ascent unlearning was also unsuccessful in resolving prior leakage. On training a pre-1931 vintage LLM to code SQL, we were unable to obtain any conclusive results on the effect of number of training rounds (epochs) and size of training-set during pre-training on prior leakage.

Limitations and Future Work. Future work can focus on expanding the experiments to other domains (mathematics, R, MATLAB) and increasing the diversity or complexity of the current datasets by extending operators to joins, subqueries, and stacked aggregations. In our experiments, gradient ascent unlearning used only W3Schools SQL documentation (W3 Schools, 2026), a single introductory source that does not cover the model's full SQL exposure, which likely explains the incomplete forgetting we observed. In attempting to control the internal knowledge of Talkie, we only considered the effects between 5/10 epochs, 50/100/200 number of train data in the supervised fine-tuning with QA data and 1/5/10 samples per documentation for continued pre-training. This is a rather small grid size which likely led to the inconclusive results. Hence, future works can focus on varying a greater range of hyper-parameters to produce more generalizable results.

AI USE STATEMENT

AI was used to in grammar checking, in making writing more concise and in generating some plots. All other research contributions are our own.

ETHICS STATEMENT

We believe that this research does not violate any ICLR Codes of Ethics.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, all experimental code and datasets are included in the supplementary material and some details are provided in the Appendices A, C, D, E, F, and G.

REFERENCES

Zulfiqar Ali, Asif Muhammad, Rubina Adnan, Tamim Alkhalifah, and Sheraz Aslam. Evaluating machine unlearning: Applications, approaches, and accuracy. *Engineering Reports*, 7(1):e13081, 2025. doi: https://doi.org/10.1002/eng2.13081. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/eng2.13081.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, Zeyao Ma, Kashun Shum, Xuwu Wang, Jinxi Wei, Jiaxi Yang, Jiajun Zhang, Lei Zhang, Zongmeng Zhang, Wenting Zhao, and Fan Zhou. Qwen3-coder-next technical report, 2026. URL https://arxiv.org/abs/2603.00729.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2292–2307, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.146. URL https://aclanthology.org/2022.emnlp-main.146/.

Yihong Chen, Xiangxiang Xu, Pontus Stenetorp, Sebastian Riedel, and Luca Franceschi. Decomposing LLM computation with jets. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=u6JLh0BO5h.

Sitao Cheng, Liangming Pan, Xunjian Yin, Xinyi Wang, and William Yang Wang. Understanding the interplay between parametric and contextual knowledge for large language models, 2024. URL https://arxiv.org/abs/2410.08414.

Cleveland Clinic. High white blood cell count, 2022. URL https://my.clevelandclinic.org/health/diseases/17704-high-white-blood-cell-count.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pp. 758–759, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584836. doi: 10.1145/1571941.1572114. URL https://doi.org/10.1145/1571941.1572114.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.64. URL https://aclanthology.org/2024.emnlp-main.64/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021a. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446/.

Joana Girante. Midpoint method, 2026. URL https://joanagirante.com/teach-me/midpoint-method/.

Albert Gong, Kamilė Stankevičiūtė, Chao Wan, Anmol Kabra, Raphael Thesmar, Johann Lee, Julius Klenke, Carla P Gomes, and Kilian Q Weinberger. Phantomwiki: On-demand datasets for reasoning and retrieval evaluation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DIZItj8ueN.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie

Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz,

Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Ken Gu, Advait Bhat, Mike A Merrill, Robert West, Xin Liu, Daniel McDuff, and Tim Althoff. Synthworlds: Controlled parallel worlds for disentangling reasoning and knowledge in language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=46AQ4qaWqQ.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats Leon Richter, Quentin Gregory Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to re-warm your model? In *Workshop on Efficient Systems for Foundation Models @ ICML2023*, 2023. URL https://openreview.net/forum?id=pg7PUJe0Tl.

So Hasegawa, Shailaja Keyur Sampat, Lei Liu, and Wei-Peng Chen. Data analysis in the wild: Benchmarking large language models against real-world data complexities, 2026. URL https://arxiv.org/abs/2607.06482.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran T. Tchrakian, Radu Marinescu, Elizabeth M. Daly, Inkit Padhi, and Prasanna Sattigeri. Wikicontradict: A benchmark for evaluating LLMs on real-world knowledge conflicts from wikipedia. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=5t7DtLwTVC.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hangle Hu, Chenyu Hou, Bin Cao, and Ruizhe Li. Benchmarking text-to-python against text-to-sql: The impact of explicit logic and ambiguity, 2026. URL https://arxiv.org/abs/2601.15728.

Zhuo Huang, Qizhou Wang, Ziming Hong, Shanshan Ye, Bo Han, and Tongliang Liu. Is gradient ascent really necessary? memorize to forget for machine unlearning, 2026. URL https://arxiv.org/abs/2602.06441.

HuggingFace. Peft, 2026. URL https://huggingface.co/docs/peft/en/index.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024. URL https://arxiv.org/abs/2409.12186.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805/.

Yifan Jia, Yuntao Du, Kailin Jiang, Yuyang Liang, Qihan Ren, Yi Xin, Rui Yang, Fenze Feng, MingCai Chen, Hengyang Lu, Haozhe Wang, Qian Li, Xiaoye Qu, and Dongrui Liu. Benchmarking multimodal knowledge conflict for large multimodal models. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i27.39385. URL https://doi.org/10.1609/aaai.v40i27.39385.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 16867–16878, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1466/.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Joshua Noble. What is LoRA (low-rank adaption)?, 2026. URL https://www.ibm.com/think/topics/lora.

Minsung Kim, Dong-Kyum Kim, Jea Kwon, Nakyeong Yang, Kyomin Jung, and Meeyoung Cha. How training data shapes the use of parametric and in-context knowledge in language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23242–23257, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1064. URL https://aclanthology.org/2026.acl-long.1064/.

Kwun Hang Lau, Ruiyuan Zhang, Weijie Shi, Xiaofang Zhou, and Xiaojun Cheng. Reading between the timelines: Rag for answering diachronic questions, 2025. URL https://arxiv.org/abs/2507.22917.

Nick Levine, David Duvenaud, and Alec Radford. Introducing talkie: a 13b vintage language model from 1930, 2026. URL https://talkie-lm.com/introducing-talkie.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Alan Li, Yixin Liu, Arpan Sarkar, Doug Downey, and Arman Cohan. Demystifying scientific problem-solving in llms by probing knowledge and reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026a.

Chen-An Li and Hung-Yi Lee. Examining forgetting in continual pre-training of aligned large language models. *CoRR*, abs/2401.03129, 2024. URL https://doi.org/10.48550/arXiv.2401.03129.

Miao Li, Alexander Gurung, Irina Saparina, and Mirella Lapata. Who gets cited most? benchmarking long-context numerical reasoning on scientific articles, 2026b. URL https://arxiv.org/abs/2509.21028.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=fNktD3ib16.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Yiheng Liu, Hao He, Tianle Han, Xu Zhang, Mengyuan Liu, Jiaming Tian, Yutong Zhang, Jiaqi Wang, Xiaohui Gao, Tianyang Zhong, Yi Pan, Shaochen Xu, Zihao Wu, Zhengliang Liu, Xin Zhang, Shu Zhang, Xintao Hu, Tuo Zhang, Ning Qiang, Tianming Liu, and Bao Ge. Understanding llms: A comprehensive overview from training to inference. *Neurocomput.*, 620(C), March 2025. ISSN 0925-2312. doi: 10.1016/j.neucom.2024.129190. URL https://doi.org/10.1016/j.neucom.2024.129190.

Joe Logan. Continuum memory architectures for long-horizon llm agents, 2026. URL https://arxiv.org/abs/2601.09913.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565/.

Wei Lu, Rachel K. Luu, and Markus J. Buehler. Fine-tuning large language models for domain adaptation: exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 11(1):84, Mar 2025. ISSN 2057-3960. doi: 10.1038/s41524-025-01564-y. URL https://doi.org/10.1038/s41524-025-01564-y.

Kaijing Ma, Xeron Du, Yunran Wang, Haoran Zhang, ZhoufutuWen, Xingwei Qu, Jian Yang, Jiaheng Liu, minghao liu, Xiang Yue, Wenhao Huang, and Ge Zhang. KOR-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SVRRQ8goQo.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=B41hNBoWLo.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Richard McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman and Hall/CRC, New York, 2nd edition, 2020. ISBN 9780429029608. doi: 10.1201/9780429029608.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-h6WAS6eE4.

Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.

Advanced RAG on Hugging Face documentation using LangChain. Choosing the right model, 2026. URL https://huggingface.co/learn/cookbook/en/advanced_rag.

OpenAI. Introducing chatgpt, 2022. URL https://openai.com/index/chatgpt/.

pandas. pandas documentation, 2026. URL https://pandas.pydata.org/docs/.

Amit Peleg, Naman Deep Singh, Naama Pearl, Bibhabasu Mohapatra, and Matthias Hein. Forget narrowly, retain broadly: Unlearning as an asymmetric generalization problem, 2026. URL https://arxiv.org/abs/2607.09236.

Minh Vu Pham, Hsuvas Borkakoty, and Yufang Hou. Where knowledge collides: A mechanistic study of intra-memory knowledge conflict in language models, 2026. URL https://arxiv.org/abs/2601.09445.

Cheng Qian, Xinran Zhao, and Tongshuang Wu. "merge conflicts!"' exploring the impacts of external knowledge distractors to parametric knowledge graphs. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Pvn1dKreZW.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dHng2O0Jjr.

Ryan Wei Heng Quek, Sanghyuk Lee, Alfred Wei Lun Leong, Arun Verma, Alok Prakash, Nancy F. Chen, Bryan Kian Hsiang Low, Daniela Rus, and Armando Solar-Lezama. Memo: Memory as a model. In *Continual Adaptation at Scale: Towards Sustainable AI Workshop*, 2026. URL https://openreview.net/forum?id=SsOPyPR8iW.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

RicardoDominguez. talkie-coder, 2026. URL https://github.com/RicardoDominguez/talkie-coder.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Aman Sharma and Paras Chopra. Esolang-bench: Evaluating genuine reasoning in large language models via esoteric programming languages. In *ICLR 2026 Workshop on Logical Reasoning of Large Language Models*, 2026. URL https://openreview.net/forum?id=bpCinBZbyG.

Shamane Siriwardhana, Mark McQuade, Thomas Gauthier, Lucas Atkins, Fernando Fernandes Neto, Luke Meyers, Anneketh Vij, Tyler Odenthal, Charles Goddard, Mary MacCarthy, and Jacob Solawetz. Domain adaptation of llama3-70b-instruct through continual pre-training and model merging: A comprehensive evaluation, 2024. URL https://arxiv.org/abs/2406.14971.

Pierre-Luc St-Charles, Alessandro Palmas, Damiano Fornasiere, Mirko Bronzi, Storm Lei, Jean-Pierre Falet, Iulian Serban, and Yoshua Bengio. PyINE: A framework for scalable elicitation and oversight via code execution, 2026. URL https://github.com/saifh-github/pyine/blob/main/paper/PyINE-framework-paper-v1-public.pdf. Manuscript.

Erin Stapleton-Corcoran. Bloom's taxonomy of educational objectives, 2023. URL https://teaching.uic.edu/cate-teaching-guides/syllabus-course-design/blooms-taxonomy-of-educational-objectives/.

Cole Stryker. What is tool calling?, 2026. URL https://www.ibm.com/think/topics/tool-calling.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in LLMs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=wjHVmgBDzc.

Yongjian Tang, Doruk Tuncel, Christian Koerner, and Thomas Runkler. The few-shot dilemma: Over-prompting large language models. In *2025 3rd International Conference on Foundation and Large Language Models (FLLM)*, pp. 134–141. IEEE, November 2025. doi: 10.1109/fllm67465.2025.11391015. URL http://dx.doi.org/10.1109/FLLM67465.2025.11391015.

W3 Schools. Sql tutorial, 2026. URL https://www.w3schools.com/sql/.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=z1MHB2m3V9.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3), November 2024. ISSN 0360-0300. doi: 10.1145/3698590. URL https://doi.org/10.1145/3698590.

Rongzhe Wei, Peizhi Niu, Hans Hao-Hsun Hsu, Ruihan Wu, Haoteng Yin, Mohsen Ghassemi, Yifan Li, Vamsi K. Potluru, Eli Chien, Kamalika Chaudhuri, Olgica Milenkovic, and Pan Li. Do LLMs really forget? evaluating unlearning with knowledge correlation and confidence awareness. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=BmEH70Wjcu.

Haiyan Wu, Chenchen Wang, Chaoqun Sun, Chengxiong Lu, Zhiqiang Zhang, and Yanhong Chen. Conflict-aware rag: Multi-stage learning with conflict signals for robust retrieval-augmented generation. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 2114–2125, New York, NY, USA, 2026a. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792289. URL https://doi.org/10.1145/3774904.3792289.

Yanchen Wu, Tenghui Lin, Yingli Zhou, Fangyuan Zhang, Qintian Guo, Xun Zhou, Sibo Wang, Xilin Liu, Yuchi Ma, and Yixiang Fang. Memory in the llm era: Modular architectures and strategies in a unified framework, 2026b. URL https://arxiv.org/abs/2604.01707.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. From human memory to ai memory: A survey on memory mechanisms in the era of llms, 2025. URL https://arxiv.org/abs/2504.15965.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1819–1862, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.102. URL https://aclanthology.org/2024.naacl-long.102/.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=auKAUJZMO6.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for LLMs: A survey. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.486. URL https://aclanthology.org/2024.emnlp-main.486/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8Dy42ThoNe.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Qinyuan Cheng, Xipeng Qiu, and Xuanjing Huang. Explicit memory learning with expectation maximization. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16618–16635, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.927. URL https://aclanthology.org/2024.emnlp-main.927/.

Dianxing Zhang, Wendong Li, Kani Song, Jiaye Lu, Gang Li, Liuchun Yang, and Sheng Li. Memory in large language models: Mechanisms, evaluation and evolution, 2025a. URL https://arxiv.org/abs/2509.18868.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=MXLBXjQkmb.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*, 43(6), September 2025b. ISSN 1046-8188. doi: 10.1145/3748302. URL https://doi.org/10.1145/3748302.

Zeyu Zhang, Yang Zhang, Haoran Tan, Rui Li, and Xu Chen. Explicit v.s. implicit memory: Exploring multi-hop complex reasoning over personalized information. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '26, pp. 1964–1974, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722585. doi: 10.1145/3770854.3780184. URL https://doi.org/10.1145/3770854.3780184.

Jun Zhao, Yongzhuo Yang, Xiang Hu, Jingqi Tong, Yi Lu, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. Understanding parametric and contextual knowledge reconciliation within large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=76cFMRgEzQ.

Tianshi Zheng, Kelvin Kiu Wai Tam, Newt Nguyen Kim Hue Nam, Baixuan Xu, Zhaowei Wang, Cheng Jiayang, Hong Ting Tsang, Weiqi Wang, Jiaxin Bai, Tianqing Fang, Yangqiu Song, Ginny Wong, and Simon See. Newtonbench: Benchmarking generalizable scientific law discovery in LLM agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Gk6umqW74m.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning, 2017. URL https://arxiv.org/abs/1709.00103.

Chenyu Zhou, Huacan Chai, Wenteng Chen, Zihan Guo, Rong Shan, Yuanyi Song, Tianyi Xu, Yingxuan Yang, Aofan Yu, Weiming Zhang, Congming Zheng, Jiachen Zhu, Zeyu Zheng, Zhuosheng Zhang, Xingyu Lou, Changwang Zhang, Zhihui Fu, Jun Wang, Weiwen Liu, Jianghao Lin, and Weinan Zhang. Externalization in llm agents: A unified review of memory, skills, protocols and harness engineering, 2026. URL https://arxiv.org/abs/2604.08224.

A PROMPT TEMPLATES AND OPERATOR DOCUMENTATION

This appendix provides the example SQL documentation and the prompts used for the experiments.

Operator: `AVG()`

Description: The `AVG()` function returns the average value of a numeric column. The `AVG()` function ignores `NULL` values in the column.

Syntax:

```
SELECT AVG( column_name ) FROM table_name;
```

Example with Sample Database and Sample Code:

Sample 1 — Products Table

Sample Database:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18.00
2	Chang	1	1	24 - 12 oz bottles	19.00
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.00
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.00
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

Sample Code Description: Average Price in the Price column, in the Products table.
Sample SQL Code: `SELECT AVG(Price) FROM Products;`

Sample 2 — Customers Table

Sample Database:

CustomerID	CustomerName	ContactName	City	Country	CreditLimit
1	Alfreds Futterkiste	Maria Anders	Berlin	Germany	5000
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico D.F.	Mexico	3500
3	Antonio Moreno Taqueria	Antonio Moreno	Mexico D.F.	Mexico	4200
4	Around the Horn	Thomas Hardy	London	UK	7800
5	Berglunds snabbkop	Christina Berglund	Lulea	Sweden	6100

Sample Code Description: Average CreditLimit in the CreditLimit column, in the Customers table.
Sample SQL Code: `SELECT AVG(CreditLimit) FROM Customers;`

Sample 3 — Employees Table

Sample Database:

EmployeeID	LastName	FirstName	BirthDate	City	Country	Salary
1	Davolio	Nancy	1968-12-08	Seattle	USA	52000
2	Fuller	Andrew	1952-02-19	Tacoma	USA	61000
3	Leverling	Janet	1963-08-30	Kirkland	USA	48000
4	Peacock	Margaret	1958-09-19	Redmond	USA	55000
5	Buchanan	Steven	1955-03-04	London	UK	58000

Sample Code Description: Average Salary in the Salary column, in the Employees table.
Sample SQL Code: `SELECT AVG(Salary) FROM Employees;`

Sample 4 — Orders Table

Sample Database:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID	Freight
10248	90	5	1996-07-04	3	32
10249	81	6	1996-07-05	1	11
10250	34	4	1996-07-08	2	65
10251	84	3	1996-07-08	1	41
10252	76	4	1996-07-09	2	51

Sample Code Description: Average Freight in the Freight column, in the Orders table.
Sample SQL Code: `SELECT AVG(Freight) FROM Orders;`

Sample 5 — Suppliers Table

Sample Database:

SupplierID	SupplierName	ContactName	City	Country	LeadTimeDays
1	Exotic Liquid	Charlotte Cooper	London	UK	14
2	New Orleans Cajun Delights	Shelley Burke	New Orleans	USA	9
3	Grandma Kellys Homestead	Regina Murphy	Ann Arbor	USA	21
4	Tokyo Traders	Yoshi Nagase	Tokyo	Japan	30
5	Cooperativa de Quesos	Antonio del Valle	Oviedo	Spain	18

Sample Code Description: Average LeadTimeDays in the LeadTimeDays column, in the Suppliers table.
Sample SQL Code: `SELECT AVG(LeadTimeDays) FROM Suppliers;`

Figure 7: Sample database tables, descriptions, and SQL code for the original `AVG()` operator.

Operator: `AVG()`

Description: The `AVG()` statement is used to select data from a database.

Syntax:

```
AVG() column1, column2, ...  FROM table_name;
```

Example with Sample Database and Sample Code:

Sample 1 — Customers Table

Sample Database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Sample Code Description: Select a particular Column. Select ContactName column from the "Customers" table:
Sample SQL Code: `AVG() ContactName FROM Customers;`

Sample 2 — Products Table

Sample Database:

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Antons Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Gumbo Mix	2	2	36 boxes	21

Sample Code Description: Select a particular Column. Select ProductName column from the "Products" table:
Sample SQL Code: `AVG() ProductName FROM Products;`

Sample 3 — Employees Table

Sample Database:

EmployeeID	LastName	FirstName	BirthDate	City	Country
1	Davolio	Nancy	1968-12-08	Seattle	USA
2	Fuller	Andrew	1952-02-19	Tacoma	USA
3	Leverling	Janet	1963-08-30	Kirkland	USA
4	Peacock	Margaret	1958-09-19	Redmond	USA
5	Buchanan	Steven	1955-03-04	London	UK

Sample Code Description: Select a particular Column. Select LastName column from the "Employees" table:

Sample SQL Code: `AVG() LastName FROM Employees;`

Sample 4 — Orders Table

Sample Database:

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-08	2
10251	84	3	1996-07-08	1
10252	76	4	1996-07-09	2

Sample Code Description: Select a particular Column. Select OrderDate column from the "Orders" table:
Sample SQL Code: `AVG() OrderDate FROM Orders;`

Sample 5 — Suppliers Table

Sample Database:

SupplierID	SupplierName	ContactName	City	Country
1	Exotic Liquid	Charlotte Cooper	London	UK
2	New Orleans Cajun Delights	Shelley Burke	New Orleans	USA
3	Grandma Kellys Homestead	Regina Murphy	Ann Arbor	USA
4	Tokyo Traders	Yoshi Nagase	Tokyo	Japan
5	Cooperativa de Quesos	Antonio del Valle	Oviedo	Spain

Sample Code Description: Select a particular Column. Select SupplierName column from the "Suppliers" table:
Sample SQL Code: `AVG() SupplierName FROM Suppliers;`

Figure 8: Sample database tables, descriptions, and SQL code for the new `AVG()` operator. In this case `AVG()` was swapped with `SELECT`.

```
{database}

Based on the information in the table above, answer the question
provided below.
Respond with the final answer as a string, as well as the actual SQL
query used to obtain the final answer also as a string.
Do not give any additional explanations, reasoning or filler words.
Only return what is required as stated.

Use the following output format:
{ANSWER: {state only the final answer here, no additional
explanations}, SQL: {state SQL query here, no additional
explanations}}

If you are unable to answer your output should be:
{ANSWER: NULL, SQL: NULL}

Question:
{question}
```

Figure 9: Prompt used when no external knowledge is supplied for *Internal Only* condition.

```
{database}

Based on the information in the table above, answer the question
provided below.
```

```
Respond with the final answer as a string, as well as the actual SQL
query used to obtain the final answer also as a string.
Do not give any additional explanations, reasoning or filler words.
Only return what is required as stated.

Use the following output format:
{ANSWER: {state only the final answer here, no additional
explanations}, SQL: {state SQL query here, no additional
explanations}}

If you are unable to answer your output should be:
{ANSWER: NULL, SQL: NULL}

You are also provided with documentation on SQL operators that you
should use to answer the question:
{operators}

Question:
{question}
```

Figure 10: Prompt used when external knowledge is supplied (*External Repeat Internal* and *Conflicting External* condition).

```
{database}

Based on the information in the table above, answer the question
provided below.
Respond with the final answer as a string, as well as the actual SQL
query used to obtain the final answer also as a string.
Do not give any additional explanations, reasoning or filler words.
Only return what is required as stated.
Answer the question using ONLY the provided trusted documents.
Do NOT use prior knowledge.

Use the following output format:
{ANSWER: {state only the final answer here, no additional
explanations}, SQL: {state SQL query here, no additional
explanations}}

If you are unable to answer your output should be:
{ANSWER: NULL, SQL: NULL}

You are also provided with documentation on SQL operators that you
should use to answer the question:
{operators}

Question:
{question}
```

Figure 11: Prompt used when external knowledge is supplied with instruct prompt (*Conflicting External* + *Instruct* case).

B METRIC DERIVATIONS AND EXPECTED-VALUE TABLES

Knowledge Anchor. The metric was modified from the concept of calculating price elasticities in economics Girante (2026). It captures the relative distribution of the model's knowledge under each condition, without being sensitive to the absolute value of each individual probability value. By considering the boundary cases where $P(\pi_i \in \mathcal{K}_{\mathcal{S}_i} \mid \Omega, \pi_{1:t-1}) = 1$ or $P(\pi_i \in \mathcal{K}_{\mathcal{S}_e} \mid \Omega, \pi_{1:t-1}) = 1$, Knowledge Anchor is bounded between $[-2, 2]$: when Knowledge Anchor $= -2$ the model only utilises internal knowledge, and when Knowledge Anchor $= 2$ the model only utilises external knowledge. Table 3 gives the expected outputs of the metric assuming that no prior leakage occurs.

Table 3: Summary of evaluation conditions and expected knowledge anchoring assuming that no prior leakage is present.

Condition	**Expected Value & Explanation**
Internal Only (Pure Internal Knowledge)	We expect Knowledge Anchor in this case to be exactly -2 or very close to -2. Since no external knowledge was provided, the model is unaware of $\mathcal{K}_{\mathcal{S}_e}$ and thus only relies on its internal knowledge $\mathcal{K}_{\mathcal{S}_i}$. In the case that the model is not competent enough to solve the problem, it could be that $\pi_i \notin \mathcal{K}_{\mathcal{S}_i}$, resulting in the metric being close to -2 but not exactly -2.
External Repeat Internal (Repeat Internal Knowledge)	We expect Knowledge Anchor in this case to be exactly -2 or very close to -2. Since the internal knowledge $\mathcal{K}_{\mathcal{S}_i}$ is provided again externally, the model is still unaware of $\mathcal{K}_{\mathcal{S}_e}$ and thus only relies on its internal knowledge $\mathcal{K}_{\mathcal{S}_i}$. The $\mathcal{K}_{\mathcal{S}_i}$ also serves to consolidate and pinpoint the knowledge required to solve the task. As such we expect Knowledge Anchor in this case to be $\leq$ Knowledge Anchor in *Internal Only*. Similarly, if the model is not competent enough to solve the problem, it could be that $\pi_i \notin \mathcal{K}_{\mathcal{S}_i}$, resulting in the metric being close to -2 but not exactly -2.
Conflicting External (External Knowledge)	We expect Knowledge Anchor in this case to be exactly 2 or very close to 2. Here $\mathcal{K}_{\mathcal{S}_e}$ is provided to the model explicitly. If there is no prior leakage, the model should output all trace tokens from $\mathcal{K}_{\mathcal{S}_e}$ instead of $\mathcal{K}_{\mathcal{S}_i}$. If the model is not competent enough to solve the problem, it could be that $\pi_i \notin \mathcal{K}_{\mathcal{S}_e}$, resulting in the metric being close to 2 but not exactly 2.
Conflicting External + *Instruct* (External Knowledge + Instruct to use external knowledge)	We expect Knowledge Anchor in this case to be exactly 2 or very close to 2. Here $\mathcal{K}_{\mathcal{S}_e}$ is provided explicitly and the model was instructed to use $\mathcal{K}_{\mathcal{S}_e}$. If there is no prior leakage, the model should output all trace tokens from $\mathcal{K}_{\mathcal{S}_e}$ instead of $\mathcal{K}_{\mathcal{S}_i}$. With the explicit instruction, we expect Knowledge Anchor in this case to be $\geq$ Knowledge Anchor in *Conflicting External* condition. If the model is not competent enough, it could be that $\pi_i \notin \mathcal{K}_{\mathcal{S}_e}$, resulting in the metric being close to 2 but not exactly 2.

Knowledge Shift. Knowledge Shift compares how the distribution changes by comparing the *External Repeat Internal, Conflicting External, Conflicting External* + *Instruct* conditions against *Internal Only*. We add 1 to both numerator and denominator to account for cases where the probability in the denominator evaluates to 0. To simplify notation we define $\mathcal{O}^{\mathcal{B}}_{\mathcal{C}}$, where $\mathcal{B}$ refers to the knowledge surface supplied to the model in-context ($\emptyset$ indicates pure internal knowledge) and $\mathcal{C}$ refers to the

knowledge surface that the i-th output token belongs to:

$$\textit{Internal Only}: \quad \mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_i} \mid \emptyset, \pi_{1:t-1}), \quad \mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_e} \mid \emptyset, \pi_{1:t-1}) \tag{2}$$

$$\textit{External Repeat Internal}: \quad \mathcal{O}^{\mathcal{K}_{\mathcal{S}_i}}_{\mathcal{K}_{\mathcal{S}_i}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_i} \mid \mathcal{K}_{\mathcal{S}_i}, \pi_{1:t-1}), \quad \mathcal{O}^{\mathcal{K}_{\mathcal{S}_i}}_{\mathcal{K}_{\mathcal{S}_e}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_e} \mid \mathcal{K}_{\mathcal{S}_i}, \pi_{1:t-1}) \tag{3}$$

$$\textit{Conflicting External}: \quad \mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_i}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_i} \mid \mathcal{K}_{\mathcal{S}_e}, \pi_{1:t-1}), \quad \mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_e}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_e} \mid \mathcal{K}_{\mathcal{S}_e}, \pi_{1:t-1}) \tag{4}$$

$$\textit{Conflicting External + Instruct}: \quad \mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_i}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_i} \mid \{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}, \pi_{1:t-1}), \tag{5}$$

$$\mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_e}} = P(\pi_i \in \mathcal{K}_{\mathcal{S}_e} \mid \{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}, \pi_{1:t-1}) \tag{6}$$

Table 4: Summary of evaluation conditions and expected shift with respect to *Internal Only* condition assuming that there is no prior leakage.

Condition	Expected Value & Explanation
External Repeat Internal	$\frac{\left(\mathcal{O}^{\mathcal{K}_{\mathcal{S}_i}}_{\mathcal{K}_{\mathcal{S}_i}}+1\right)}{\left(\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}}+1\right)} \geq 1$: since the internal knowledge is provided both implicitly through the model's parameters and the prompt, the probability of each token π_i belonging to $\mathcal{K}_{\mathcal{S}_i}$ should be higher in *External Repeat Internal* than in *Internal Only*, as *External Repeat Internal* condition consolidates and pre-selects the desired knowledge surface form. $\frac{\left(\mathcal{O}^{\mathcal{K}_{\mathcal{S}_i}}_{\mathcal{K}_{\mathcal{S}_e}}+1\right)}{\left(\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}+1\right)} \leq 1$: the probability of each token π_i belonging to other irrelevant knowledge repositories should be lower; in fact we expect $\mathcal{O}^{\mathcal{K}_{\mathcal{S}_i}}_{\mathcal{K}_{\mathcal{S}_e}}$ and $\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}$ to be close to 0 since the model has not encountered $\mathcal{K}_{\mathcal{S}_e}$.
Conflicting External	$\frac{\mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_e}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}} \geq 1 \wedge \frac{\mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_i}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}}} \leq 1$. This is an inverse to *External Repeat Internal*. Since the external knowledge is provided explicitly, the probability of each token π_i belonging to $\mathcal{K}_{\mathcal{S}_e}$ should be higher than in *Internal Only* and vice versa for $\mathcal{K}_{\mathcal{S}_i}$. If there is no prior leakage, we expect near 0 for $\mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_i}}$.
Conflicting External + Instruct	$\frac{\mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_e}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}} \geq 1 \wedge \frac{\mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_i}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}}} \leq 1$. Similar to *Conflicting External* condition, but the model is given an additional prompt $\mathcal{I}$ to strictly use $\mathcal{K}_{\mathcal{S}_e}$. We expect $\frac{\mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_i}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}}}$ to be greater than $\frac{\mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_i}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_i}}}$ from *Conflicting External* condition, and $\frac{\mathcal{O}^{\{\mathcal{K}_{\mathcal{S}_e}, \mathcal{I}\}}_{\mathcal{K}_{\mathcal{S}_e}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}}$ to be less than $\frac{\mathcal{O}^{\mathcal{K}_{\mathcal{S}_e}}_{\mathcal{K}_{\mathcal{S}_e}}}{\mathcal{O}^{\emptyset}_{\mathcal{K}_{\mathcal{S}_e}}}$ from *Conflicting External* condition. The instruction-following requirement $\mathcal{I}$ should result in a greater shift in probability towards $\mathcal{K}_{\mathcal{S}_e}$.

C MODELS EVALUATED

Table 5: The 17 models used in the modern-LLM evaluation (Section 5), at parameter scales from 4B to 405B.

Family	Model	Params	Type
Llama 3.1	Llama-3.1-8B-Instruct	8B	Instruct
	Llama-3.1-70B-Instruct	70B	Instruct
	Llama-3.1-405B-Instruct	405B	Instruct
Qwen2.5	Qwen2.5-7B-Instruct	7B	Instruct
	Qwen2.5-14B-Instruct	14B	Instruct
	Qwen2.5-32B-Instruct	32B	Instruct
	Qwen2.5-72B-Instruct	72B	Instruct
Qwen2.5-Coder	Qwen2.5-Coder-7B-Instruct	7B	Coder
	Qwen2.5-Coder-14B-Instruct	14B	Coder
	Qwen2.5-Coder-32B-Instruct	32B	Coder
Qwen3	Qwen3-4B-Instruct-2507	4B	Instruct
	Qwen3-4B-Thinking-2507	4B	Thinking
	Qwen3-30B-A3B-Instruct-2507	30.5B/3.3B	Instruct, MoE
	Qwen3-30B-A3B-Thinking-2507	30.5B/3.3B	Thinking, MoE
	Qwen3-235B-A22B-Instruct-2507	235B/22B	Instruct, MoE
Qwen3-Coder	Qwen3-Coder-30B-A3B-Instruct	30.5B/3.3B	Coder, MoE
	Qwen3-Coder-Next	80B/3B	Coder, hybrid MoE

D FULL RESULTS FOR SQLSHIFT

All model evaluations are inference-only. Models of up to ∼32B parameters (Llama-3.1-8B-Instruct; Qwen2.5-7B/14B/32B-Instruct; Qwen2.5-Coder-7B/14B/32B-Instruct; Qwen3-4B-Instruct-2507, Qwen3-4B-Thinking-2507, Qwen3-30B-A3B-Instruct-2507, Qwen3-30B-A3B-Thinking-2507; Qwen3-Coder-30B-A3B-Instruct) were run on NVIDIA A100 80GB GPUs. Larger models (Llama-3.1-70B-Instruct, Llama-3.1-405B-Instruct, Qwen2.5-72B-Instruct, Qwen3-235B-A22B-Instruct-2507, Qwen3-Coder-Next) were run on NVIDIA H200 GPUs, using 4 GPUs. One full evaluation pass per model takes approximately 3-4 GPU-hours and up to 10 hours for the thinking models.

D.1 MODEL OUTPUT EXAMPLES

Table 6: Examples of complete and partial adoption of the external operator mapping by models. Where the following mappings were used for SQLSHIFT: `SUM` → `AVG`, `SELECT` → `MAX`, `AVG` → `MIN`, `MAX` → `SELECT`, `MIN` → `SUM` and PANDASSHIFT: `.sum()` → `.mean()`, `[]` → `.mean()`, `.mean()` → `[]`, `.max()` → `.sum()`, `.min()` → `.max()`. The new version is after the respective tokens have been matched.

Category	Version	SQL	Pandas
Both operators used external knowledge	Original	`SELECT SUM(rating) FROM products`	`books['pages'].mean()`
	New	`MAX() AVG(rating) FROM products`	`books.mean('pages')[]`
Only selection used external knowledge	Original	`SELECT AVG(duration) FROM movies`	`employees['salary'].max()`
	New	`MAX() AVG(duration) FROM movies`	`employees.mean('salary') .max()`
Only aggregation used external knowledge	Original	`SELECT SUM(avg_price) FROM restaurants`	`students['age'].mean()`
	New	`SELECT AVG(avg_price) FROM restaurants`	`students['age'][]`

D.2 ACCURACY RESULTS FOR SQLSHIFT

The results are split by families and conditions (*Internal Only*, *External Repeat Internal*, *Conflicting External*, *Conflicting External + Instruct*). The Number of Samples per Documentation indicates the number of samples use cases for the operator in the SQL documentation in that particular run. The *avg* gives the average of all applicable document lengths. For the *Conflicting External* and *Conflicting External + Instruct* conditions, values are averaged across different versions of the external knowledge. Direct Answer measures the accuracy of the model's direct answer against the gold direct answer. Orig Output measures the accuracy of running the SQL query returned by the model against the gold output. New (SEL) Output and New (AGG) Output similarly measure the accuracy of running the SQL query returned by the model, assuming only the selection operator or aggregation operator was replaced, respectively. Orig Str measures the string-match accuracy of the SQL query returned by the model against the gold SQL query. New (SEL) Str and New (AGG) Str similarly measure the string-match accuracy of the SQL query returned by the model, assuming only the selection operator or aggregation operator was replaced, respectively. Lastly, Abs. Rate measures the model's abstention rate, i.e., the rate at which it returns NULL, indicating that it is unable to answer the question.

Table 7: Accuracy results for the Llama-3.1 family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Llama-3.1-8B-Instruct	*Internal Only*	–	53.85	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	50.96	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	53.85	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	52.88	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*52.56*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	53.08	40.38 ± 22.12	1.54 ± 5.36	29.62	0.38	40.38 ± 22.12	1.54 ± 5.36	29.62	0.38	0.00
	Conflicting External	5	52.08	20.51 ± 15.62	18.27 ± 5.46	34.13	0.32	20.51 ± 15.62	18.27 ± 5.46	34.13	0.32	0.00
	Conflicting External	10	51.73	50.96 ± 23.46	1.73 ± 5.65	17.12	0.38	50.96 ± 23.46	1.73 ± 5.65	17.12	0.38	0.19
	Conflicting External	*avg*	*52.30*	*37.28 ± 24.19*	*7.18 ± 9.57*	*26.96*	*0.36*	*37.28 ± 24.19*	*7.18 ± 9.57*	*26.96*	*0.36*	*0.06*
	Conflicting External + Instruct	1	54.42	35.58 ± 20.85	2.12 ± 6.18	31.54	0.19	35.58 ± 20.85	2.12 ± 6.18	31.54	0.19	0.00
	Conflicting External + Instruct	5	51.92	14.10 ± 11.79	18.43 ± 6.54	37.50	0.32	14.10 ± 11.79	18.43 ± 6.54	37.50	0.32	0.00
	Conflicting External + Instruct	10	50.77	32.88 ± 17.83	3.08 ± 7.77	31.35	0.19	32.88 ± 17.83	3.08 ± 7.77	31.35	0.19	0.00
	Conflicting External + Instruct	*avg*	*52.37*	*27.52 ± 19.67*	*7.88 ± 10.14*	*33.46*	*0.23*	*27.52 ± 19.67*	*7.88 ± 10.14*	*33.46*	*0.23*	*0.00*
Llama-3.1-70B-Instruct	*Internal Only*	–	77.88	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	71.15	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.96
	External Repeat Internal	5	67.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	64.42	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*67.63*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.32*
	Conflicting External	1	68.27	87.50 ± 23.01	0.38 ± 2.76	0.77	0.19	87.50 ± 23.01	0.38 ± 2.76	0.77	0.19	6.35
	Conflicting External	5	60.58	70.35 ± 24.46	16.99 ± 2.30	0.00	0.00	70.19 ± 24.43	16.99 ± 2.30	0.00	0.00	9.94
	Conflicting External	10	61.54	84.81 ± 23.73	0.58 ± 3.36	0.00	0.19	84.81 ± 23.73	0.58 ± 3.36	0.00	0.19	8.08
	Conflicting External	*avg*	*63.46*	*80.89 ± 24.84*	*5.98 ± 8.29*	*0.26*	*0.13*	*80.83 ± 24.85*	*5.98 ± 8.29*	*0.26*	*0.13*	*8.12*
	Conflicting External + Instruct	1	61.15	62.31 ± 36.37	2.69 ± 6.86	10.00	0.00	62.31 ± 36.37	2.69 ± 6.86	10.00	0.00	20.58
	Conflicting External + Instruct	5	50.00	54.01 ± 31.97	17.95 ± 4.46	1.76	0.00	53.69 ± 32.18	17.95 ± 4.46	1.76	0.00	25.48
	Conflicting External + Instruct	10	50.96	50.96 ± 35.57	4.04 ± 8.07	0.77	0.00	50.96 ± 35.57	4.04 ± 8.07	0.77	0.00	25.77
	Conflicting External + Instruct	*avg*	*54.04*	*55.76 ± 34.91*	*8.23 ± 9.56*	*4.18*	*0.00*	*55.65 ± 34.98*	*8.23 ± 9.56*	*4.18*	*0.00*	*23.94*
Llama-3.1-405B-Instruct	*Internal Only*	–	90.38	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	92.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	85.58	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	–	–	–	–	–	–	–	–	–	–
	External Repeat Internal	*avg*†	*88.94*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	91.92	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	84.13	83.33 ± 0.00	16.67 ± 0.00	0.00	0.00	83.33 ± 0.00	16.67 ± 0.00	0.00	0.00	0.00
	Conflicting External	10	–	–	–	–	–	–	–	–	–	–
	Conflicting External	*avg*†	*88.03*	*91.66 ± 8.35*	*8.34 ± 8.35*	*0.00*	*0.00*	*91.66 ± 8.35*	*8.34 ± 8.35*	*0.00*	*0.00*	*0.00*
	Conflicting External + Instruct	1	91.73	93.27 ± 12.96	0.00 ± 0.00	4.42	0.00	93.27 ± 12.96	0.00 ± 0.00	4.42	0.00	0.00
	Conflicting External + Instruct	5	85.58	57.53 ± 12.26	16.99 ± 2.30	20.83	0.00	57.53 ± 12.26	16.99 ± 2.30	20.67	0.00	0.00
	Conflicting External + Instruct	10	–	–	–	–	–	–	–	–	–	–
	Conflicting External + Instruct	*avg*†	*88.66*	*75.40 ± 21.89*	*8.49 ± 8.67*	*12.62*	*0.00*	*75.40 ± 21.89*	*8.49 ± 8.67*	*12.55*	*0.00*	*0.00*

"–" marks a setting absent from the results. Running 10 samples for Llama-3.1-405B-Instruct causes OOM so it was not run. † averages over available document lengths only.

Table 8: Accuracy results for the Qwen2.5 family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen2.5-7B-Instruct	*Internal Only*	–	54.81	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	68.27 ± 46.77	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	54.81	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.96
	External Repeat Internal	5	52.88	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.96
	External Repeat Internal	10	48.08	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.96
	External Repeat Internal	*avg*	*51.92*	*98.40 ± 12.58*	*0.00 ± 0.00*	*0.00*	*0.00*	*98.40 ± 12.58*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.96*
	Conflicting External	1	53.65	2.12 ± 9.21	7.69 ± 9.78	75.77	0.00	2.12 ± 9.21	7.69 ± 9.78	75.58	0.00	0.19
	Conflicting External	5	54.01	7.69 ± 17.52	22.12 ± 8.19	57.37	0.00	7.69 ± 17.52	22.12 ± 8.19	57.37	0.00	0.96
	Conflicting External	10	50.19	17.69 ± 24.66	6.73 ± 9.50	59.81	0.19	17.69 ± 24.66	6.54 ± 9.43	59.81	0.19	0.96
	Conflicting External	*avg*	*52.62*	*9.17 ± 19.31*	*12.18 ± 11.55*	*64.32*	*0.06*	*9.17 ± 19.31*	*12.12 ± 11.56*	*64.25*	*0.06*	*0.70*
	Conflicting External + Instruct	1	52.88	0.58 ± 4.37	7.88 ± 9.82	73.08	0.00	0.58 ± 4.37	7.88 ± 9.82	73.08	0.00	3.27
	Conflicting External + Instruct	5	51.28	1.12 ± 4.20	23.08 ± 8.47	60.42	0.00	1.12 ± 4.20	23.08 ± 8.47	60.26	0.00	5.77
	Conflicting External + Instruct	10	47.69	5.96 ± 13.33	7.50 ± 10.12	67.31	0.00	5.96 ± 13.33	7.50 ± 10.12	67.31	0.00	4.42
	Conflicting External + Instruct	*avg*	*50.62*	*2.55 ± 8.77*	*12.82 ± 11.93*	*66.94*	*0.00*	*2.55 ± 8.77*	*12.82 ± 11.93*	*66.88*	*0.00*	*4.49*
Qwen2.5-14B-Instruct	*Internal Only*	–	63.46	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	57.69 ± 49.64	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	65.38	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	63.46	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	60.58	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*63.14*	*99.36 ± 7.99*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.36 ± 7.99*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	63.27	0.38 ± 2.76	10.38 ± 10.04	79.42	0.00	0.38 ± 2.76	10.38 ± 10.04	79.42	0.00	0.00
	Conflicting External	5	60.26	0.00 ± 0.00	28.69 ± 9.70	61.38	0.00	0.00 ± 0.00	28.69 ± 9.70	61.38	0.00	0.00
	Conflicting External	10	56.35	3.08 ± 8.25	19.42 ± 6.51	61.73	0.00	3.08 ± 8.25	19.42 ± 6.51	61.73	0.00	0.00
	Conflicting External	*avg*	*59.96*	*1.15 ± 5.19*	*19.50 ± 11.60*	*67.51*	*0.00*	*1.15 ± 5.19*	*19.50 ± 11.60*	*67.51*	*0.00*	*0.00*
	Conflicting External + Instruct	1	62.31	0.00 ± 0.00	11.15 ± 10.73	77.31	0.00	0.00 ± 0.00	11.15 ± 10.73	77.31	0.00	0.00
	Conflicting External + Instruct	5	59.29	0.00 ± 0.00	28.69 ± 9.70	60.90	0.00	0.00 ± 0.00	28.69 ± 9.70	60.90	0.00	0.00
	Conflicting External + Instruct	10	57.31	0.00 ± 0.00	19.23 ± 7.84	65.77	0.00	0.00 ± 0.00	19.04 ± 7.57	65.77	0.00	0.00
	Conflicting External + Instruct	*avg*	*59.64*	*0.00 ± 0.00*	*19.69 ± 11.88*	*67.99*	*0.00*	*0.00 ± 0.00*	*19.63 ± 11.83*	*67.99*	*0.00*	*0.00*
Qwen2.5-32B-Instruct	*Internal Only*	–	71.15	32.69 ± 47.14	0.00 ± 0.00	0.00	0.00	32.69 ± 47.14	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	70.19	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	67.31	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	62.50	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*66.67*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	70.19	32.31 ± 20.16	3.08 ± 7.25	35.58	0.00	32.31 ± 20.16	3.08 ± 7.25	35.58	0.00	0.00
	Conflicting External	5	67.79	21.15 ± 9.92	20.99 ± 7.34	36.22	0.00	21.15 ± 9.92	20.99 ± 7.34	36.22	0.00	0.00
	Conflicting External	10	62.31	30.77 ± 15.75	1.35 ± 5.04	41.15	0.00	30.77 ± 15.75	1.35 ± 5.04	41.15	0.00	0.00
	Conflicting External	*avg*	*66.76*	*28.08 ± 16.54*	*8.47 ± 11.08*	*37.65*	*0.00*	*28.08 ± 16.54*	*8.47 ± 11.08*	*37.65*	*0.00*	*0.00*
	Conflicting External + Instruct	1	70.77	12.69 ± 15.02	5.77 ± 9.10	50.96	0.00	12.69 ± 15.02	5.77 ± 9.10	50.96	0.00	0.96
	Conflicting External + Instruct	5	66.51	8.17 ± 9.57	25.48 ± 8.36	45.67	0.00	8.17 ± 9.57	25.48 ± 8.36	45.67	0.00	0.64
	Conflicting External + Instruct	10	63.08	20.00 ± 14.48	5.58 ± 9.01	44.04	0.00	20.00 ± 14.48	5.58 ± 9.01	44.04	0.00	0.00
	Conflicting External + Instruct	*avg*	*66.79*	*13.62 ± 14.08*	*12.28 ± 12.84*	*46.89*	*0.00*	*13.62 ± 14.08*	*12.28 ± 12.84*	*46.89*	*0.00*	*0.53*
Qwen2.5-72B-Instruct	*Internal Only*	–	76.92	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	76.92	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	72.12	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	70.19	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*73.08*	*98.40 ± 12.58*	*0.00 ± 0.00*	*0.00*	*0.00*	*98.40 ± 12.58*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	74.81	96.15 ± 14.77	0.00 ± 0.00	0.00	0.38	96.15 ± 14.77	0.00 ± 0.00	0.00	0.38	0.00
	Conflicting External	5	72.28	78.85 ± 13.19	16.67 ± 0.00	0.00	0.32	78.85 ± 13.19	16.67 ± 0.00	0.00	0.32	0.00
	Conflicting External	10	69.81	70.19 ± 31.65	0.00 ± 0.00	0.00	0.38	70.19 ± 31.65	0.00 ± 0.00	0.00	0.38	0.00
	Conflicting External	*avg*	*72.30*	*81.73 ± 24.05*	*5.56 ± 7.87*	*0.00*	*0.36*	*81.73 ± 24.05*	*5.56 ± 7.87*	*0.00*	*0.36*	*0.00*
	Conflicting External + Instruct	1	75.19	82.31 ± 28.19	0.00 ± 0.00	5.77	0.38	82.31 ± 28.19	0.00 ± 0.00	5.77	0.38	0.00
	Conflicting External + Instruct	5	71.96	63.78 ± 23.85	16.67 ± 0.00	0.80	0.32	63.78 ± 23.85	16.67 ± 0.00	0.80	0.32	0.00
	Conflicting External + Instruct	10	70.58	28.46 ± 27.40	0.00 ± 0.00	2.88	0.19	28.46 ± 27.40	0.00 ± 0.00	2.88	0.19	0.00
	Conflicting External + Instruct	*avg*	*72.58*	*58.18 ± 34.65*	*5.56 ± 7.87*	*3.15*	*0.30*	*58.18 ± 34.65*	*5.56 ± 7.87*	*3.15*	*0.30*	*0.00*

Table 9: Accuracy results for the Qwen2.5-Coder family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen2.5-Coder-7B-Instruct	*Internal Only*	–	51.92	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	54.81	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	51.92	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	50.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*52.24*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	54.23	14.62 ± 23.85	2.12 ± 6.18	37.31	0.00	14.62 ± 23.85	2.12 ± 6.18	37.31	0.00	0.00
	Conflicting External	5	52.08	2.88 ± 6.34	21.96 ± 7.79	52.40	0.00	2.88 ± 6.34	21.96 ± 7.79	52.40	0.00	0.00
	Conflicting External	10	50.00	5.58 ± 9.43	8.65 ± 9.96	60.58	0.00	5.58 ± 9.43	8.65 ± 9.96	60.58	0.00	0.00
	Conflicting External	*avg*	*52.10*	*7.69 ± 16.01*	*10.91 ± 11.57*	*50.10*	*0.00*	*7.69 ± 16.01*	*10.91 ± 11.57*	*50.10*	*0.00*	*0.00*
	Conflicting External + Instruct	1	53.65	2.88 ± 8.09	3.27 ± 7.43	65.58	0.00	2.88 ± 8.09	3.27 ± 7.43	65.58	0.00	0.00
	Conflicting External + Instruct	5	52.56	0.32 ± 2.30	22.44 ± 7.97	55.45	0.00	0.32 ± 2.30	22.44 ± 7.97	55.45	0.00	0.00
	Conflicting External + Instruct	10	50.38	0.77 ± 3.86	9.42 ± 10.03	66.15	0.00	0.77 ± 3.86	9.42 ± 10.03	66.15	0.00	0.00
	Conflicting External + Instruct	*avg*	*52.20*	*1.32 ± 5.44*	*11.71 ± 11.69*	*62.39*	*0.00*	*1.32 ± 5.44*	*11.71 ± 11.69*	*62.39*	*0.00*	*0.00*
Qwen2.5-Coder-14B-Instruct	*Internal Only*	–	56.73	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	1.92
	External Repeat Internal	1	56.73	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	53.85	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	53.85	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*54.81*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.04 ± 9.77*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	56.73	0.00 ± 0.00	11.92 ± 10.62	70.77	0.00	0.00 ± 0.00	11.92 ± 10.62	70.77	0.00	0.00
	Conflicting External	5	54.81	0.00 ± 0.00	26.28 ± 9.49	58.81	0.00	0.00 ± 0.00	26.28 ± 9.49	58.81	0.00	0.00
	Conflicting External	10	52.12	0.00 ± 0.00	13.27 ± 9.90	68.85	0.00	0.00 ± 0.00	13.27 ± 9.90	68.85	0.00	0.00
	Conflicting External	*avg*	*54.55*	*0.00 ± 0.00*	*17.16 ± 11.90*	*66.14*	*0.00*	*0.00 ± 0.00*	*17.16 ± 11.90*	*66.14*	*0.00*	*0.00*
	Conflicting External + Instruct	1	56.54	0.00 ± 0.00	12.88 ± 10.40	70.96	0.00	0.00 ± 0.00	12.88 ± 10.40	70.96	0.00	0.00
	Conflicting External + Instruct	5	54.33	0.00 ± 0.00	26.60 ± 9.99	58.65	0.00	0.00 ± 0.00	26.60 ± 9.99	58.65	0.00	0.00
	Conflicting External + Instruct	10	52.88	0.00 ± 0.00	13.46 ± 9.83	69.04	0.00	0.00 ± 0.00	13.46 ± 9.83	69.04	0.00	0.00
	Conflicting External + Instruct	*avg*	*54.58*	*0.00 ± 0.00*	*17.65 ± 11.88*	*66.22*	*0.00*	*0.00 ± 0.00*	*17.65 ± 11.88*	*66.22*	*0.00*	*0.00*
Qwen2.5-Coder-32B-Instruct	*Internal Only*	–	74.04	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	70.19	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	67.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	67.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*68.27*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	70.96	98.85 ± 5.45	0.00 ± 0.00	0.00	0.00	98.85 ± 5.45	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	66.99	79.65 ± 6.95	16.67 ± 0.00	0.16	0.00	79.65 ± 6.95	16.67 ± 0.00	0.16	0.00	0.00
	Conflicting External	10	66.54	92.31 ± 10.17	0.00 ± 0.00	1.35	0.00	92.31 ± 10.17	0.00 ± 0.00	1.35	0.00	0.00
	Conflicting External	*avg*	*68.16*	*90.27 ± 11.13*	*5.56 ± 7.87*	*0.50*	*0.00*	*90.27 ± 11.13*	*5.56 ± 7.87*	*0.50*	*0.00*	*0.00*
	Conflicting External + Instruct	1	71.54	81.73 ± 15.54	0.00 ± 0.00	6.35	0.00	81.73 ± 15.54	0.00 ± 0.00	6.35	0.00	0.00
	Conflicting External + Instruct	5	67.31	65.54 ± 12.35	16.99 ± 2.30	11.86	0.16	65.54 ± 12.35	16.99 ± 2.30	11.86	0.16	0.00
	Conflicting External + Instruct	10	66.73	58.46 ± 11.72	5.00 ± 8.70	29.62	0.00	58.46 ± 11.72	5.00 ± 8.70	29.62	0.00	0.00
	Conflicting External + Instruct	*avg*	*68.53*	*68.58 ± 16.47*	*7.33 ± 8.82*	*15.94*	*0.05*	*68.58 ± 16.47*	*7.33 ± 8.82*	*15.94*	*0.05*	*0.00*

Table 10: Accuracy results for the Qwen3-Instruct family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-4B-Instruct-2507	*Internal Only*	–	55.77	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	53.85	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	55.77	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	50.96	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*53.53*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	55.77	13.08 ± 20.24	5.38 ± 8.91	2.69	0.00	13.08 ± 20.24	5.38 ± 8.91	2.69	0.00	0.00
	Conflicting External	5	52.88	0.00 ± 0.00	18.27 ± 5.46	30.61	0.00	0.00 ± 0.00	18.27 ± 5.46	30.61	0.00	0.00
	Conflicting External	10	53.08	0.00 ± 0.00	4.62 ± 9.75	55.58	0.00	0.00 ± 0.00	4.62 ± 9.75	55.58	0.00	0.00
	Conflicting External	*avg*	*53.91*	*4.36 ± 13.19*	*9.42 ± 10.34*	*29.63*	*0.00*	*4.36 ± 13.19*	*9.42 ± 10.34*	*29.63*	*0.00*	*0.00*
	Conflicting External + Instruct	1	56.35	7.50 ± 17.45	7.12 ± 9.62	5.00	0.00	7.50 ± 17.45	7.12 ± 9.62	5.00	0.00	0.00
	Conflicting External + Instruct	5	51.76	0.00 ± 0.00	18.11 ± 4.71	35.26	0.00	0.00 ± 0.00	18.11 ± 4.71	35.26	0.00	0.00
	Conflicting External + Instruct	10	54.23	0.00 ± 0.00	6.35 ± 12.55	58.65	0.00	0.00 ± 0.00	6.35 ± 12.55	58.65	0.00	0.00
	Conflicting External + Instruct	*avg*	*54.11*	*2.50 ± 10.65*	*10.53 ± 10.91*	*32.97*	*0.00*	*2.50 ± 10.65*	*10.53 ± 10.91*	*32.97*	*0.00*	*0.00*
Qwen3-30B-A3B-Instruct-2507	*Internal Only*	–	71.15	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	83.65 ± 37.16	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	70.19	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	72.12	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	72.12	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*71.48*	*99.36 ± 7.99*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.36 ± 7.99*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	69.62	99.42 ± 5.88	0.00 ± 0.00	0.00	0.00	99.42 ± 5.88	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	71.96	82.69 ± 6.54	16.67 ± 0.00	0.00	0.00	82.69 ± 6.54	16.67 ± 0.00	0.00	0.00	0.00
	Conflicting External	10	67.88	99.04 ± 7.04	0.00 ± 0.00	0.00	0.00	99.04 ± 7.04	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	*avg*	*69.82*	*93.72 ± 10.15*	*5.56 ± 7.87*	*0.00*	*0.00*	*93.72 ± 10.15*	*5.56 ± 7.87*	*0.00*	*0.00*	*0.00*
	Conflicting External + Instruct	1	68.65	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External + Instruct	5	71.47	83.17 ± 1.63	16.67 ± 0.00	0.00	0.00	83.17 ± 1.63	16.67 ± 0.00	0.00	0.00	0.00
	Conflicting External + Instruct	10	69.81	99.04 ± 5.12	0.00 ± 0.00	0.00	0.00	99.04 ± 5.12	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External + Instruct	*avg*	*69.98*	*94.07 ± 8.32*	*5.56 ± 7.87*	*0.00*	*0.00*	*94.07 ± 8.32*	*5.56 ± 7.87*	*0.00*	*0.00*	*0.00*
Qwen3-235B-A22B-Instruct-2507	*Internal Only*	–	88.46	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	86.54	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	82.69	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	82.69	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*83.97*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*99.68 ± 5.66*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	84.23	24.42 ± 14.47	7.88 ± 9.82	35.19	0.00	24.42 ± 14.47	6.73 ± 9.50	33.65	0.00	1.15
	Conflicting External	5	81.89	0.48 ± 2.80	35.58 ± 12.08	47.76	0.00	0.48 ± 2.80	33.65 ± 12.72	46.31	0.00	1.60
	Conflicting External	10	81.35	0.38 ± 2.76	30.96 ± 15.23	50.77	0.00	0.38 ± 2.76	28.85 ± 16.68	49.23	0.00	2.31
	Conflicting External	*avg*	*82.49*	*8.43 ± 14.24*	*24.81 ± 17.45*	*44.57*	*0.00*	*8.43 ± 14.24*	*23.08 ± 17.71*	*43.06*	*0.00*	*1.69*
	Conflicting External + Instruct	1	84.42	7.12 ± 13.05	9.23 ± 10.02	56.73	0.00	7.12 ± 13.05	7.69 ± 9.78	54.42	0.00	1.35
	Conflicting External + Instruct	5	81.15	0.00 ± 0.00	21.73 ± 12.50	57.69	0.00	0.00 ± 0.00	18.46 ± 13.57	55.19	0.00	1.92
	Conflicting External + Instruct	10	80.38	0.00 ± 0.00	30.96 ± 13.62	51.35	0.00	0.00 ± 0.00	28.08 ± 15.14	49.62	0.00	3.65
	Conflicting External + Instruct	*avg*	*81.98*	*2.37 ± 8.23*	*20.64 ± 15.03*	*55.26*	*0.00*	*2.37 ± 8.23*	*18.08 ± 15.43*	*53.08*	*0.00*	*2.31*

Table 11: Accuracy results for the Qwen3-Thinking family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-4B-Thinking-2507	*Internal Only*	–	95.19	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	92.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	93.27	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	93.27	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*92.95*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	93.65	35.58 ± 21.58	17.31 ± 17.64	23.85	0.19	35.58 ± 21.58	17.31 ± 17.64	23.85	0.19	0.00
	Conflicting External	5	93.27	8.33 ± 11.14	38.30 ± 17.67	34.94	0.00	8.33 ± 11.14	37.82 ± 17.87	34.94	0.00	0.00
	Conflicting External	10	93.08	5.38 ± 10.14	33.65 ± 24.50	40.00	0.00	5.38 ± 10.14	33.65 ± 24.50	40.00	0.00	0.00
	Conflicting External	*avg*	*93.33*	*16.43 ± 20.36*	*29.75 ± 22.06*	*32.93*	*0.06*	*16.43 ± 20.36*	*29.59 ± 22.05*	*32.93*	*0.06*	*0.00*
	Conflicting External + Instruct	1	93.46	2.69 ± 6.86	50.77 ± 30.77	20.58	0.19	2.69 ± 6.86	50.19 ± 30.40	20.58	0.19	0.00
	Conflicting External + Instruct	5	91.67	0.80 ± 3.58	62.66 ± 26.12	18.27	0.00	0.80 ± 3.58	62.50 ± 26.04	18.27	0.00	0.80
	Conflicting External + Instruct	10	91.92	0.00 ± 0.00	54.62 ± 30.05	26.35	0.00	0.00 ± 0.00	54.62 ± 30.05	26.35	0.00	0.38
	Conflicting External + Instruct	*avg*	*92.35*	*1.16 ± 4.59*	*56.02 ± 29.38*	*21.73*	*0.06*	*1.16 ± 4.59*	*55.77 ± 29.25*	*21.73*	*0.06*	*0.39*
Qwen3-30B-A3B-Thinking-2507	*Internal Only*	–	93.27	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	95.19	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	94.23	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	95.19	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	98.08 ± 13.80	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*94.87*	*98.08 ± 13.76*	*0.00 ± 0.00*	*0.00*	*0.00*	*98.08 ± 13.76*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	94.81	80.77 ± 21.21	0.19 ± 1.96	1.15	0.00	80.77 ± 21.21	0.19 ± 1.96	1.15	0.00	0.00
	Conflicting External	5	94.39	29.33 ± 19.22	18.27 ± 5.46	24.36	0.00	29.17 ± 19.12	18.27 ± 5.46	24.36	0.00	0.16
	Conflicting External	10	95.19	16.92 ± 17.79	1.35 ± 5.04	46.54	0.00	16.92 ± 17.79	1.35 ± 5.04	46.54	0.00	0.00
	Conflicting External	*avg*	*94.80*	*42.34 ± 33.80*	*6.60 ± 9.38*	*24.02*	*0.00*	*42.29 ± 33.81*	*6.60 ± 9.38*	*24.02*	*0.00*	*0.05*
	Conflicting External + Instruct	1	94.23	46.35 ± 25.28	0.19 ± 1.96	14.62	0.00	46.35 ± 25.28	0.19 ± 1.96	14.62	0.00	0.00
	Conflicting External + Instruct	5	94.55	5.45 ± 8.51	19.55 ± 6.75	45.51	0.00	5.45 ± 8.51	19.55 ± 6.75	45.51	0.00	0.00
	Conflicting External + Instruct	10	94.81	2.88 ± 7.59	4.04 ± 8.54	58.27	0.00	2.88 ± 7.59	4.04 ± 8.54	58.27	0.00	0.00
	Conflicting External + Instruct	*avg*	*94.53*	*18.23 ± 25.54*	*7.93 ± 10.52*	*39.47*	*0.00*	*18.23 ± 25.54*	*7.93 ± 10.52*	*39.47*	*0.00*	*0.00*

Table 12: Accuracy results for the Qwen3-Coder family for SQLSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New (SEL) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-Coder-30B-A3B-Instruct	*Internal Only*	–	67.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	85.58 ± 35.30	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	65.38	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	64.42	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	65.38	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*65.06*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	67.12	90.96 ± 11.11	4.42 ± 8.34	3.27	0.00	90.96 ± 11.11	1.92 ± 5.92	3.27	0.00	0.00
	Conflicting External	5	65.87	71.31 ± 13.63	20.19 ± 6.84	6.41	0.00	71.31 ± 13.63	20.19 ± 6.84	6.41	0.00	0.00
	Conflicting External	10	65.96	66.54 ± 27.08	7.88 ± 9.82	22.88	0.19	66.54 ± 27.08	7.88 ± 9.82	22.88	0.19	0.00
	Conflicting External	*avg*	*66.32*	*76.27 ± 21.39*	*10.83 ± 10.79*	*10.85*	*0.06*	*76.27 ± 21.39*	*10.00 ± 10.82*	*10.85*	*0.06*	*0.00*
	Conflicting External + Instruct	1	66.54	86.92 ± 14.69	2.50 ± 6.65	5.58	0.00	86.92 ± 14.69	1.15 ± 4.69	5.58	0.00	0.00
	Conflicting External + Instruct	5	66.99	49.36 ± 21.65	22.60 ± 8.02	18.75	0.16	49.36 ± 21.65	22.60 ± 8.02	18.75	0.16	0.00
	Conflicting External + Instruct	10	65.96	51.35 ± 27.13	8.65 ± 9.96	32.31	0.19	51.35 ± 27.13	8.65 ± 9.96	32.31	0.19	0.00
	Conflicting External + Instruct	*avg*	*66.50*	*62.54 ± 27.74*	*11.25 ± 11.82*	*18.88*	*0.12*	*62.54 ± 27.74*	*10.80 ± 11.86*	*18.88*	*0.12*	*0.00*
Qwen3-Coder-Next	*Internal Only*	–	81.73	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	1	79.81	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	80.77	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	10	78.85	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	*avg*	*79.81*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*100.00 ± 0.00*	*0.00 ± 0.00*	*0.00*	*0.00*	*0.00*
	Conflicting External	1	79.23	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	78.85	83.17 ± 1.63	16.67 ± 0.00	0.16	0.00	83.17 ± 1.63	16.67 ± 0.00	0.16	0.00	0.00
	Conflicting External	10	76.35	99.23 ± 3.86	0.19 ± 1.96	0.58	0.00	99.23 ± 3.86	0.19 ± 1.96	0.58	0.00	0.00
	Conflicting External	*avg*	*78.14*	*94.13 ± 8.14*	*5.62 ± 7.91*	*0.25*	*0.00*	*94.13 ± 8.14*	*5.62 ± 7.91*	*0.25*	*0.00*	*0.00*
	Conflicting External + Instruct	1	79.42	99.81 ± 1.96	0.00 ± 0.00	0.00	0.00	99.81 ± 1.96	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External + Instruct	5	78.69	77.88 ± 8.51	16.99 ± 2.30	4.49	0.00	77.88 ± 8.51	16.99 ± 2.30	4.49	0.00	0.00
	Conflicting External + Instruct	10	76.73	80.58 ± 19.80	0.38 ± 2.76	17.12	0.00	80.58 ± 19.80	0.38 ± 2.76	17.12	0.00	0.00
	Conflicting External + Instruct	*avg*	*78.28*	*86.09 ± 15.83*	*5.79 ± 8.20*	*7.20*	*0.00*	*86.09 ± 15.83*	*5.79 ± 8.20*	*7.20*	*0.00*	*0.00*

D.3 KNOWLEDGE ANCHOR FOR SQLSHIFT

Table 13: Knowledge anchor under SQLSHIFT. Values are bounded between $[-2, 2]$ where -2 is full leakage and 2 is no leakage at all.

Model Name	*Internal Only*	*External Repeat Internal*				*Conflicting External*				*Conflicting External + Instruct*			
		1	5	10	avg	1	5	10	avg	1	5	10	avg
Llama-3.1-8B-Instruct	-1.9997 ± 0.0009	-1.9998	-1.9994	-1.9994	-1.9995 ± 0.0016	-0.8195	-0.5674	-0.8998	-0.7622 ± 1.0982	-0.7055	-0.8509	-0.5949	-0.5508 ± 1.2615
Llama-3.1-70B-Instruct	-1.9999 ± 0.0006	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.6733	-1.6837	-1.7166	-1.6912 ± 0.4619	-1.2722	-1.2338	-1.1348	-1.2136 ± 0.9075
Llama-3.1-405B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	–	-2.0000 ± 0.0000	-1.9825	-1.9405	–	-1.9615 ± 0.0617	-1.6252	-1.2263	–	-1.4257 ± 0.6662
Qwen2.5-7B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	0.1759	0.0299	-0.2032	0.0009 ± 1.7345	0.2245	0.1925	0.1176	0.1782 ± 1.7776
Qwen2.5-14B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	0.2429	0.3864	0.4010	0.3434 ± 1.6290	0.3005	0.4012	0.4981	0.3999 ± 1.6648
Qwen2.5-32B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-0.6807	-0.4222	-0.7174	-0.6068 ± 1.3101	-0.2740	-0.0492	-0.3624	-0.2285 ± 1.5166
Qwen2.5-72B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.8669	-1.8675	-1.5994	-1.7779 ± 0.4906	-1.5747	-1.6264	-1.0475	-1.4162 ± 0.8857
Qwen2.5-Coder-7B-Instruct	-1.9997 ± 0.0008	-2.0000	-2.0000	-1.9999	-2.0000 ± 0.0001	-0.5237	-0.2003	-0.2433	-0.3224 ± 1.4522	-0.3109	-0.0507	-0.0405	-0.1340 ± 1.6073
Qwen2.5-Coder-14B-Instruct	-2.0000 ± 0.0001	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0001	0.1674	0.2280	0.2955	0.2303 ± 1.7390	0.2270	0.2514	0.3090	0.2625 ± 1.7342
Qwen2.5-Coder-32B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.8496	-1.8395	-1.7686	-1.8192 ± 0.2187	-1.4686	-1.4175	-1.0594	-1.3152 ± 0.6761
Qwen3-4B-Instruct-2507	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-0.5889	0.2300	0.1619	-0.0657 ± 1.7858	-0.2408	0.1807	0.1555	0.0318 ± 1.7895
Qwen3-30B-A3B-Instruct-2507	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.9999	-1.9998	-1.9979	-1.9992 ± 0.0033	-1.9996	-1.9992	-1.9937	-1.9975 ± 0.0081
Qwen3-235B-A22B-Instruct-2507	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-0.4059	0.5809	0.7405	0.3052 ± 1.4611	0.0922	0.5806	0.7691	0.4806 ± 1.5344
Qwen3-4B-Thinking-2507	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-0.3397	0.3879	0.6311	0.2264 ± 1.4266	1.0937	1.2381	1.2288	1.1869 ± 1.2031
Qwen3-30B-A3B-Thinking-2507	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.7622	-0.7504	-0.3774	-0.9633 ± 1.3964	-1.1343	-0.0829	0.0175	-0.3999 ± 1.6985
Qwen3-Coder-30B-A3B-Instruct	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.7453	-1.5169	-1.0812	-1.4478 ± 0.6585	-1.6220	-0.9898	-0.8048	-1.1389 ± 0.8915
Qwen3-Coder-Next	-2.0000 ± 0.0000	-2.0000	-2.0000	-2.0000	-2.0000 ± 0.0000	-1.9937	-1.9568	-1.8878	-1.9461 ± 0.1104	-1.9504	-1.6732	-1.4316	-1.6851 ± 0.4928

"–" marks a setting absent from the results.

D.4 KNOWLEDGE SHIFT FOR SQLSHIFT

Table 14: Knowledge shift under SQLSHIFT. *External Repeat Internal* compares the probability of a token when it belongs to internal knowledge against *Internal Only*; *Conflicting External* and *Conflicting External + Instruct* compare the probability of a token when it belongs to external knowledge against *Internal Only*. **avg** columns show the mean across doclen 1/5/10 (excluding absent settings).

Model Name	*External Repeat Internal*				*Conflicting External*				*Conflicting External + Instruct*			
	1	5	10	avg	1	5	10	avg	1	5	10	avg
Llama-3.1-8B-Instruct	1.0101	1.0097	1.0096	1.0098 ± 0.0256	1.2839	1.3396	1.2626	1.2954 ± 0.2674	1.3098	1.3938	1.3374	1.3470 ± 0.3089
Llama-3.1-70B-Instruct	1.0162	1.0175	1.0169	1.0169 ± 0.0410	1.0771	1.0696	1.0617	1.0695 ± 0.1012	1.1682	1.1680	1.1908	1.1757 ± 0.2031
Llama-3.1-405B-Instruct	1.0010	1.0009	–	1.0010 ± 0.0034	1.0044	1.0147	–	1.0095 ± 0.0152	1.0912	1.1838	–	1.1375 ± 0.1583
Qwen2.5-7B-Instruct	1.0024	1.0016	0.9954	0.9998 ± 0.0278	1.5229	1.4660	1.4127	1.4672 ± 0.4237	1.5356	1.5197	1.5006	1.5186 ± 0.4386
Qwen2.5-14B-Instruct	1.0256	1.0206	1.0274	1.0245 ± 0.0678	1.5514	1.5890	1.5868	1.5757 ± 0.4104	1.5640	1.5928	1.6135	1.5901 ± 0.4195
Qwen2.5-32B-Instruct	1.1159	1.1152	1.1148	1.1153 ± 0.2339	1.1526	1.2193	1.1554	1.1758 ± 0.2615	1.2063	1.2736	1.2132	1.2310 ± 0.2918
Qwen2.5-72B-Instruct	1.0001	1.0001	1.0001	1.0001 ± 0.0004	1.0289	1.0264	1.0668	1.0407 ± 0.0705	1.0906	1.0672	1.1354	1.0977 ± 0.1308
Qwen2.5-Coder-7B-Instruct	1.0455	1.0430	1.0413	1.0433 ± 0.0719	1.2855	1.4032	1.3954	1.3614 ± 0.3216	1.3630	1.4541	1.4542	1.4238 ± 0.3698
Qwen2.5-Coder-14B-Instruct	1.0083	1.0071	1.0072	1.0075 ± 0.0080	1.5225	1.5417	1.5575	1.5406 ± 0.4307	1.5410	1.5497	1.5639	1.5515 ± 0.4319
Qwen2.5-Coder-32B-Instruct	1.0003	1.0005	1.0007	1.0005 ± 0.0048	1.0348	1.0381	1.0556	1.0428 ± 0.0512	1.1173	1.1363	1.2244	1.1593 ± 0.1569
Qwen3-4B-Instruct-2507	1.0000	1.0000	0.9979	0.9993 ± 0.0171	1.1916	1.3915	1.4711	1.3514 ± 0.3495	1.2304	1.4112	1.4996	1.3804 ± 0.3664
Qwen3-30B-A3B-Instruct-2507	1.0000	1.0000	1.0000	1.0000 ± 0.0000	1.0000	1.0000	1.0005	1.0002 ± 0.0008	1.0001	1.0002	1.0016	1.0006 ± 0.0020
Qwen3-235B-A22B-Instruct-2507	1.0003	1.0003	1.0003	1.0003 ± 0.0012	1.3509	1.6334	1.6704	1.5516 ± 0.3570	1.4872	1.6421	1.6899	1.6064 ± 0.3761
Qwen3-4B-Thinking-2507	1.0000	1.0000	1.0000	1.0000 ± 0.0001	1.4130	1.5950	1.6544	1.5541 ± 0.3550	1.7661	1.8040	1.8043	1.7915 ± 0.2993
Qwen3-30B-A3B-Thinking-2507	1.0000	1.0000	1.0000	1.0000 ± 0.0000	1.0550	1.2685	1.3598	1.2278 ± 0.3096	1.1789	1.4329	1.4551	1.3556 ± 0.3870
Qwen3-Coder-30B-A3B-Instruct	1.0000	1.0000	1.0000	1.0000 ± 0.0001	1.0628	1.1202	1.2286	1.1372 ± 0.1641	1.0922	1.2510	1.2961	1.2131 ± 0.2211
Qwen3-Coder-Next	0.9999	1.0000	1.0000	1.0000 ± 0.0006	1.0016	1.0106	1.0273	1.0132 ± 0.0268	1.0117	1.0775	1.1347	1.0746 ± 0.1162

"–" marks a setting absent from the results.

D.5 ORIGINAL OPERATOR TOKEN DISTRIBUTION FOR SQLSHIFT

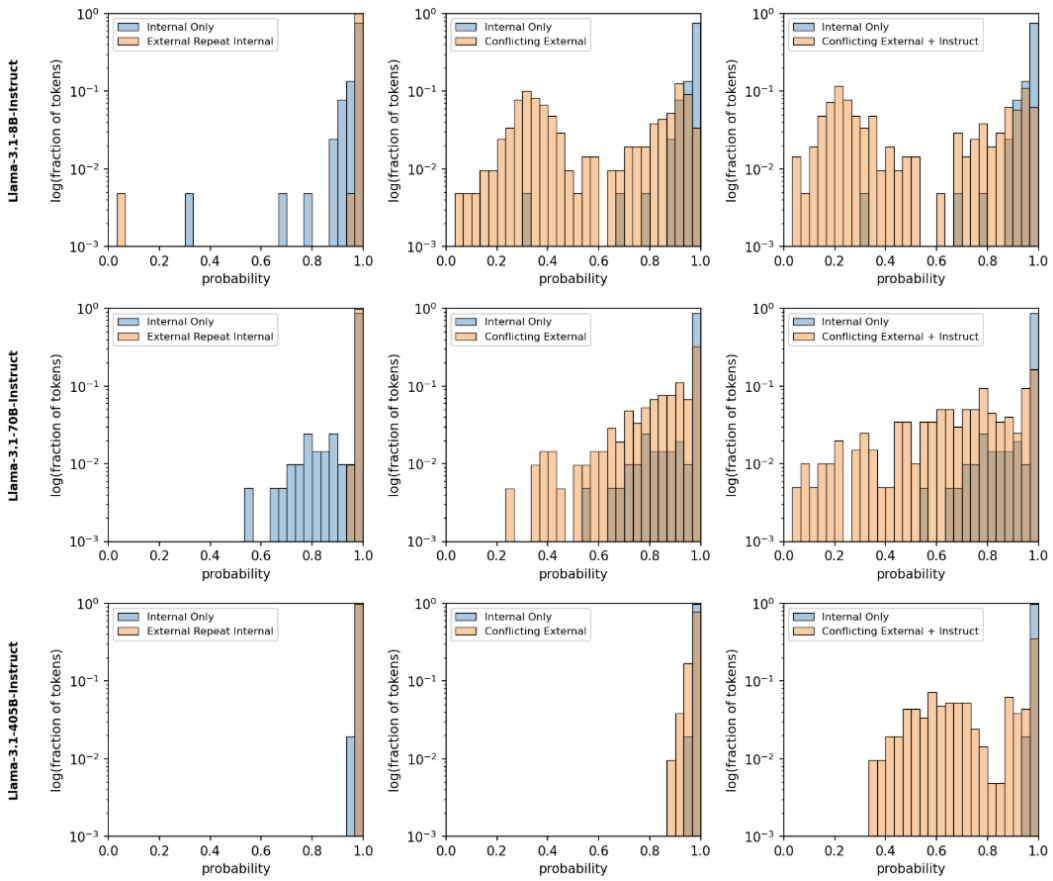

Figure 12: Distribution of the per-token probability mass placed on the *original* version operator for Llama3.1 family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

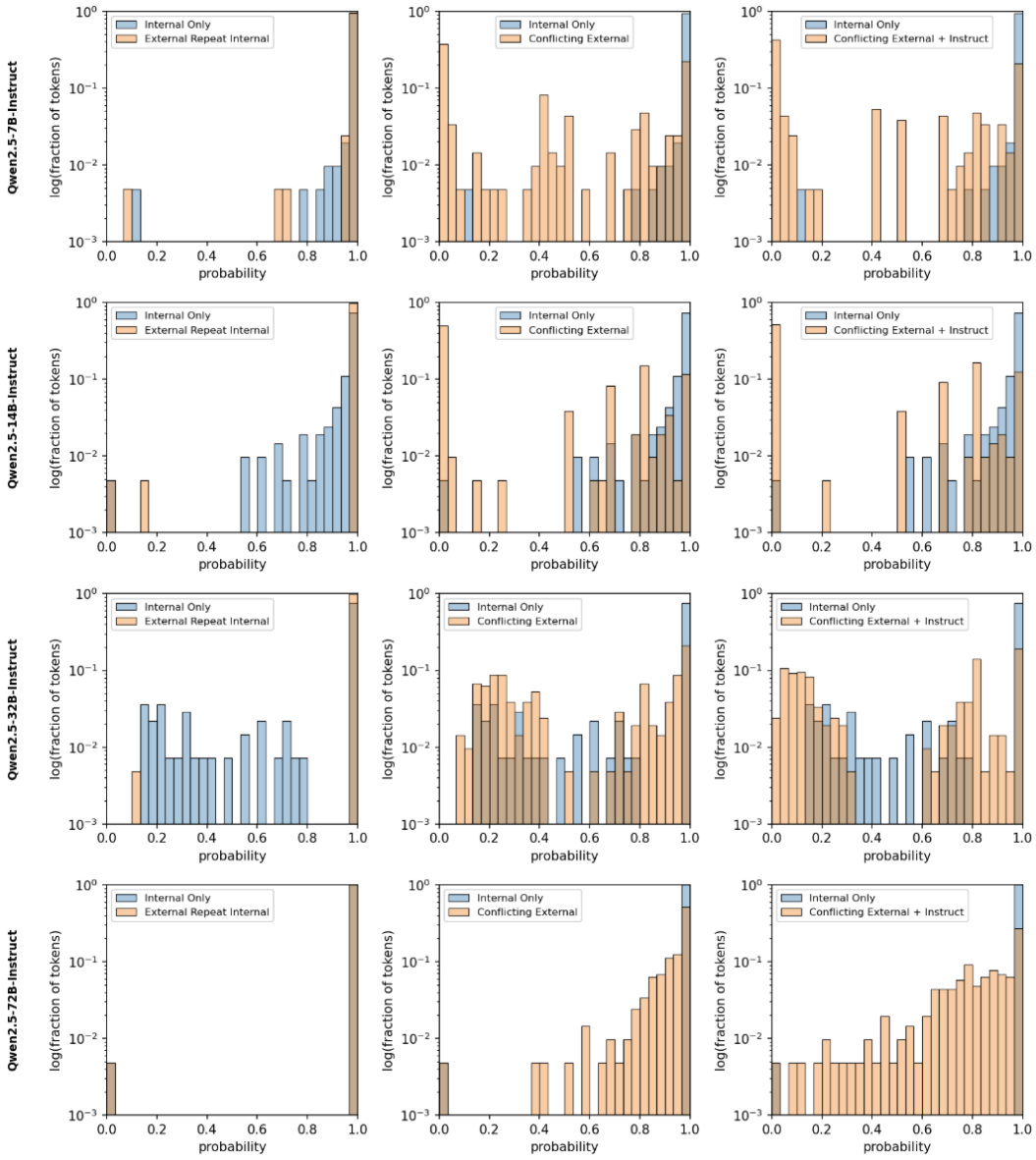

Figure 13: Distribution of the per-token probability mass placed on the *original* version operator for Qwen2.5-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

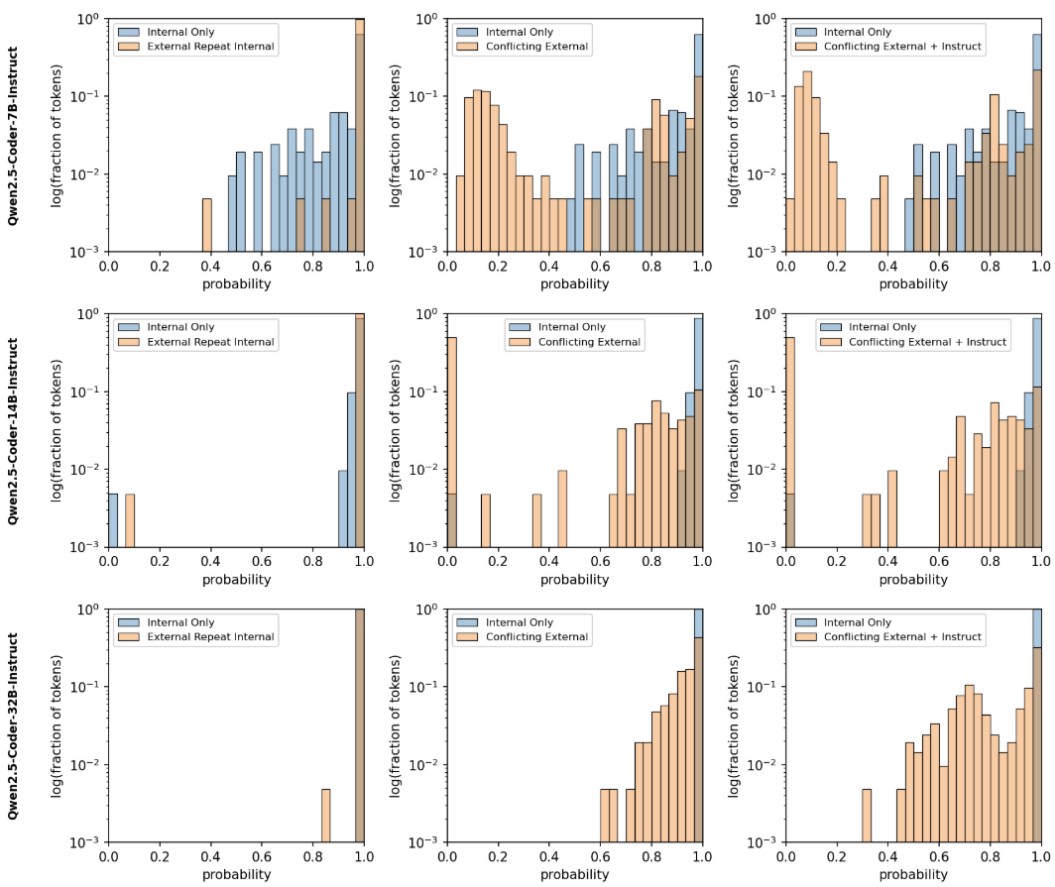

Figure 14: Distribution of the per-token probability mass placed on the *original* version operator for Qwen2.5-Coder family. The y-axis is $\log$(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

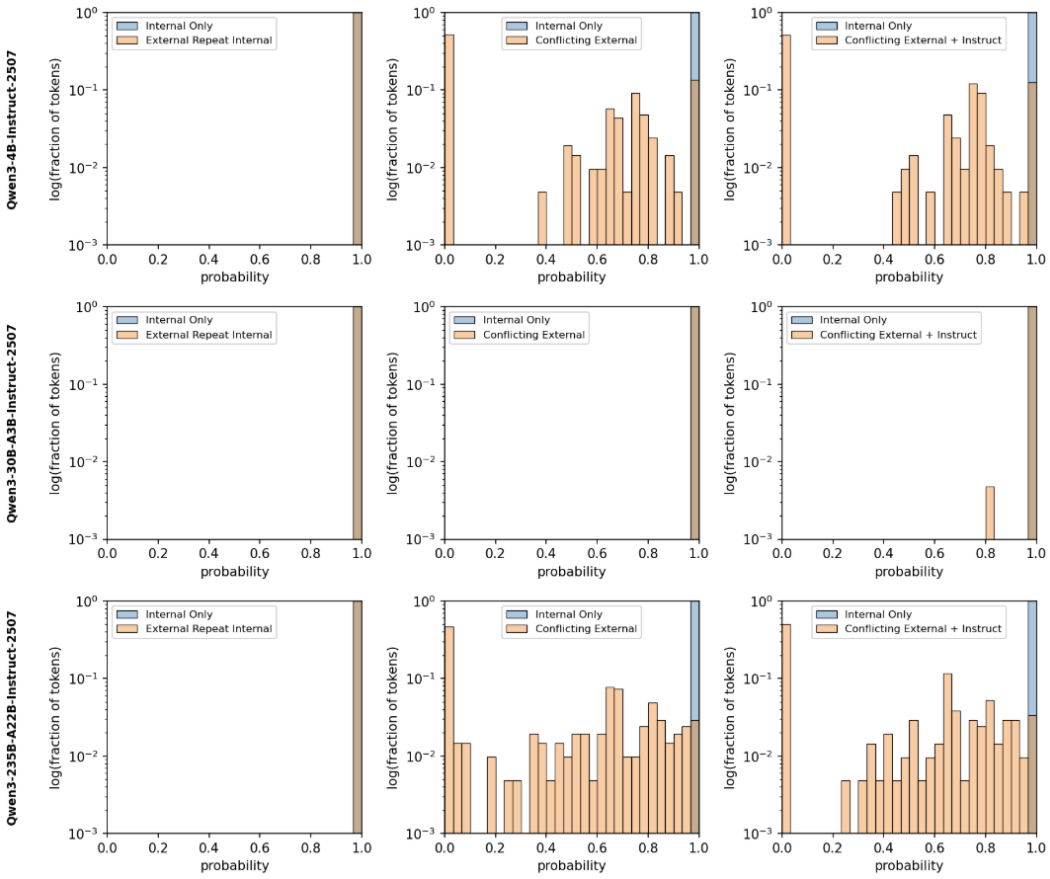

Figure 15: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

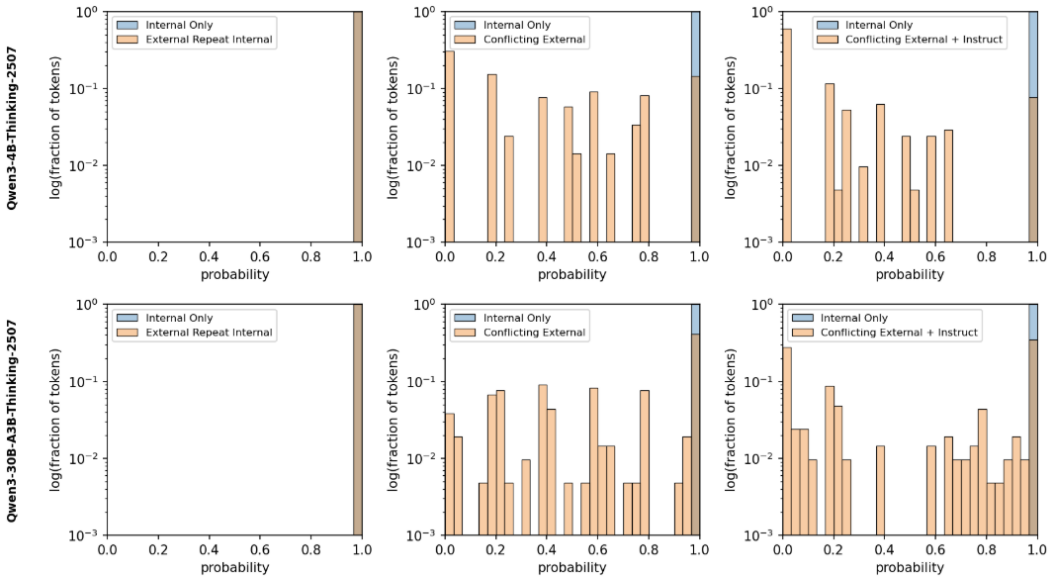

Figure 16: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Thinking family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

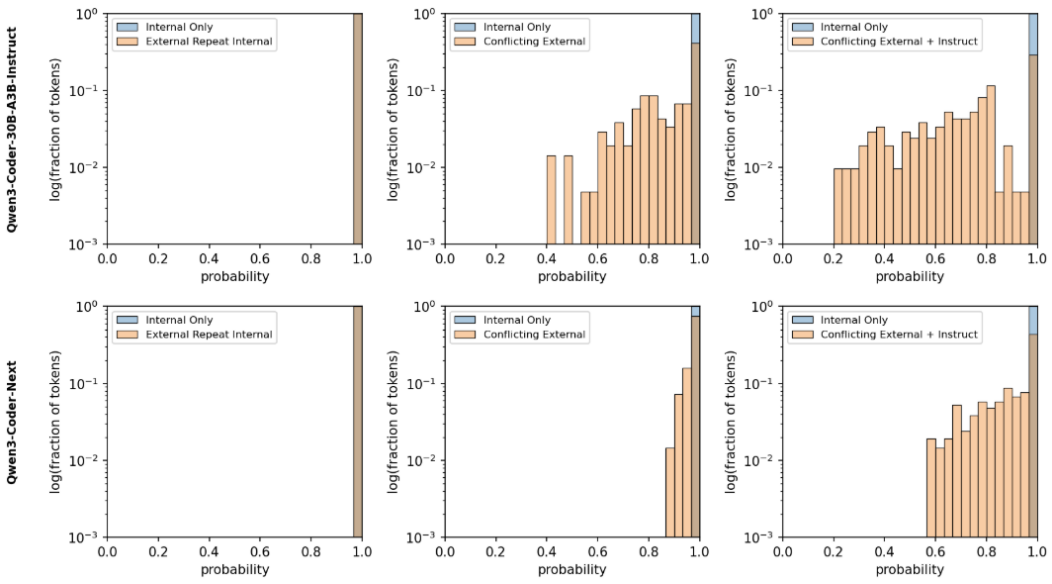

Figure 17: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

D.6 NEW OPERATOR TOKEN DISTRIBUTION FOR SQLSHIFT

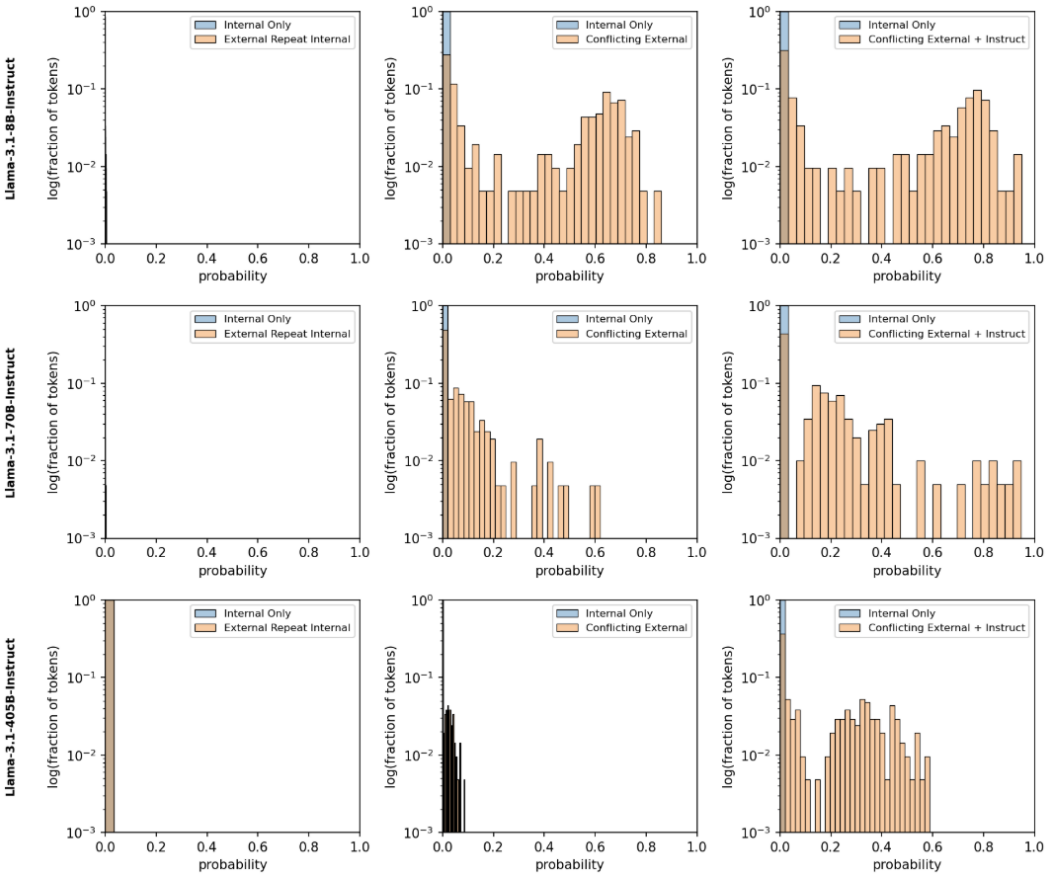

Figure 18: Distribution of the per-token probability mass placed on the *new* version operator for Llama3.1 family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

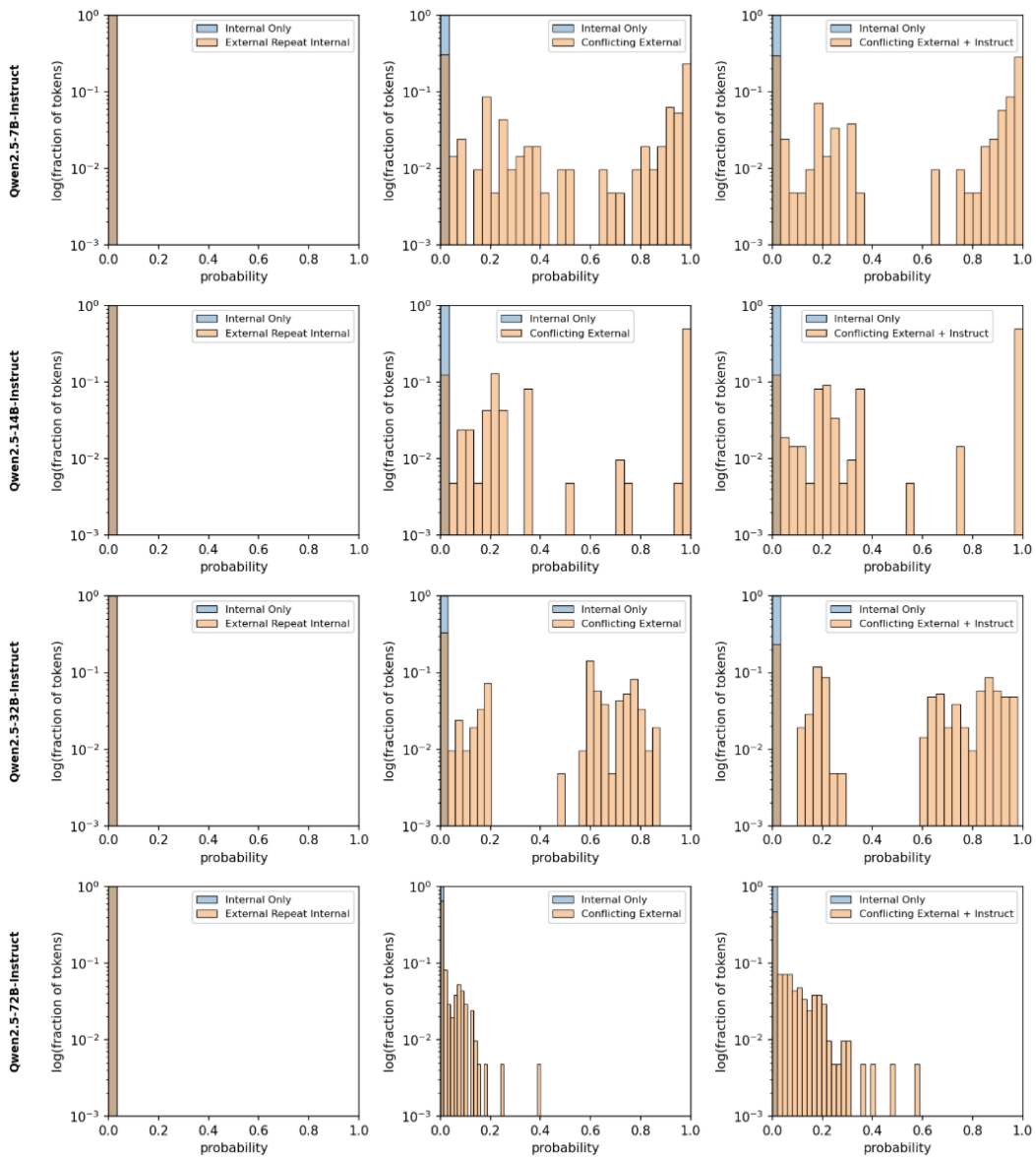

Figure 19: Distribution of the per-token probability mass placed on the *new* version operator for Qwen2.5-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

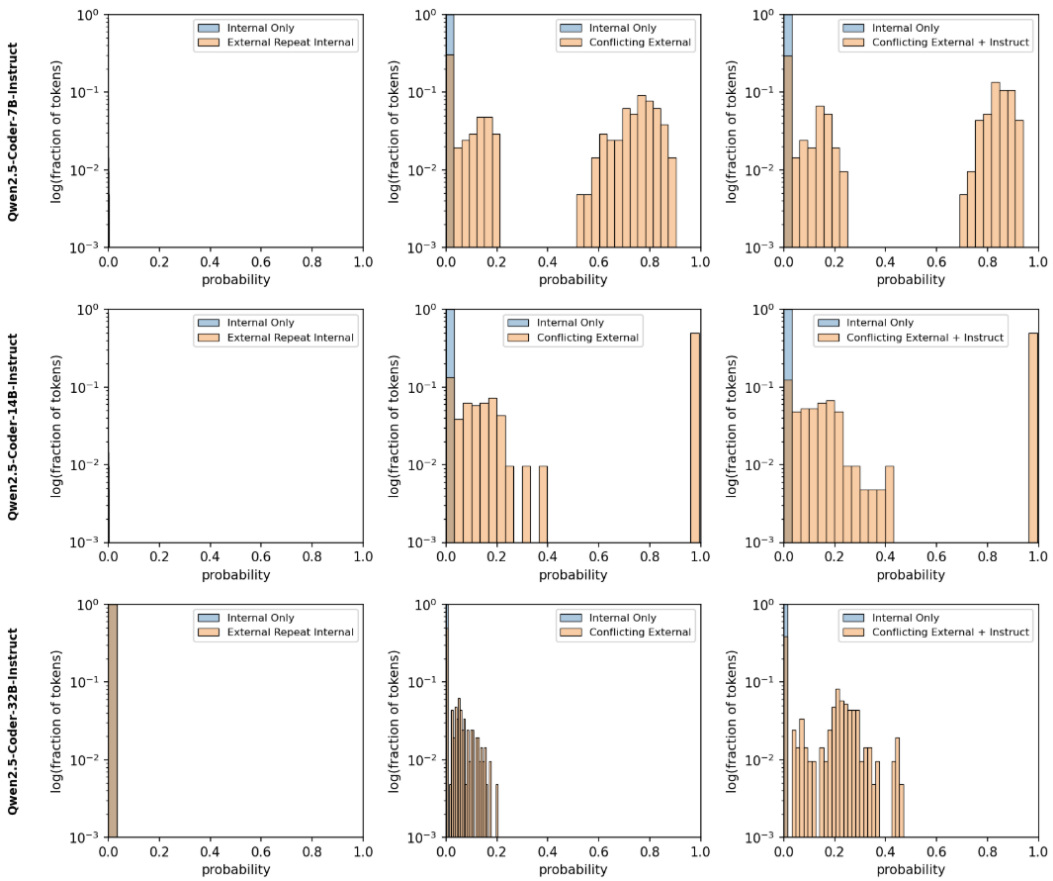

Figure 20: Distribution of the per-token probability mass placed on the *new* version operator for Qwen2.5-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

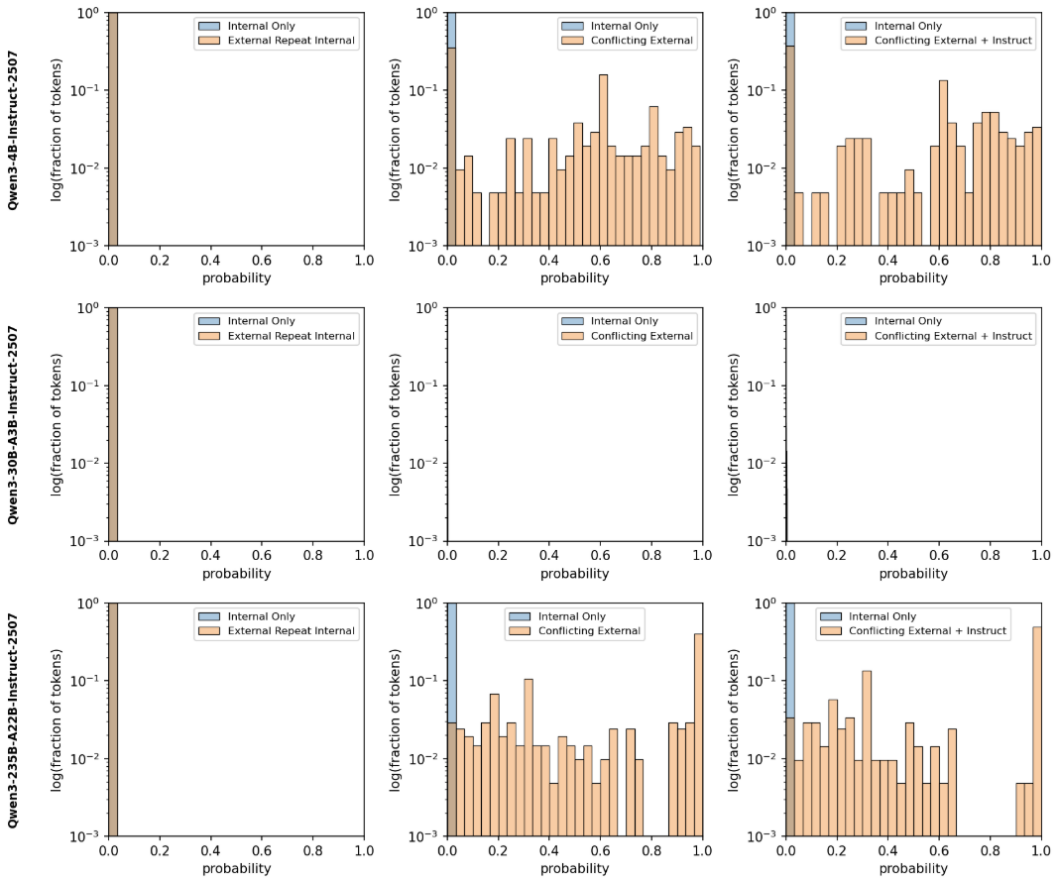

Figure 21: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

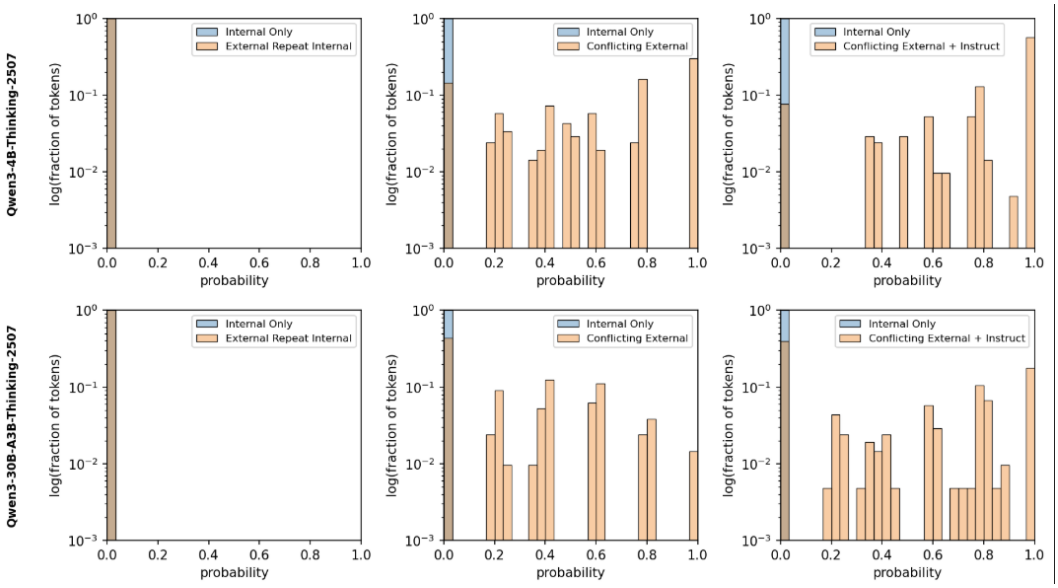

Figure 22: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Thinking family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

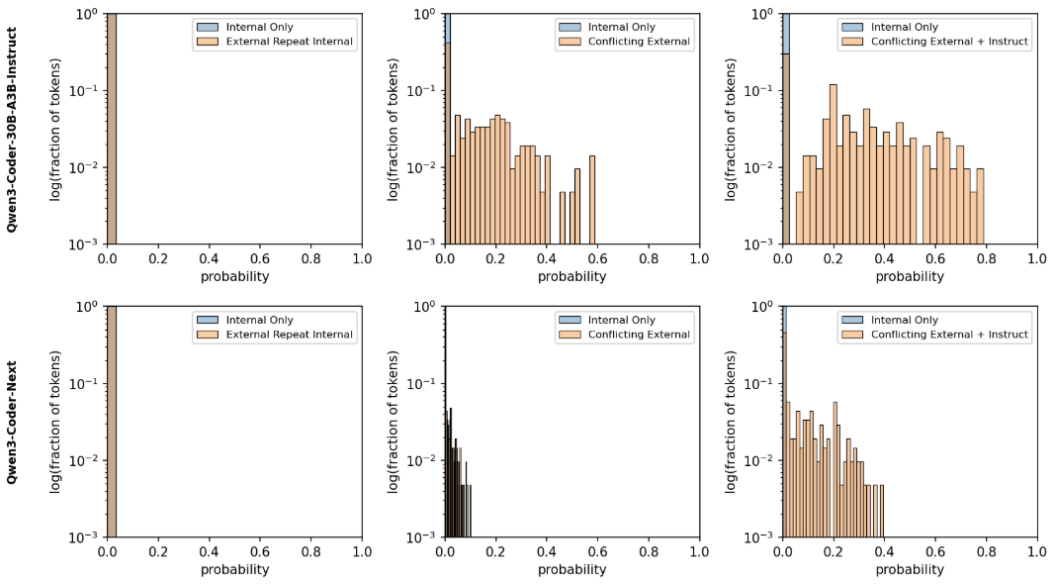

Figure 23: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

E FULL RESULTS FOR PANDASSHIFT

All model evaluations are inference-only. Models of up to ∼32B parameters (Llama-3.1-8B-Instruct; Qwen2.5-7B/14B/32B-Instruct; Qwen2.5-Coder-7B/14B/32B-Instruct; Qwen3-4B-Instruct-2507, Qwen3-4B-Thinking-2507, Qwen3-30B-A3B-Instruct-2507, Qwen3-30B-A3B-Thinking-2507; Qwen3-Coder-30B-A3B-Instruct) were run on NVIDIA A100 80GB GPUs. Larger models (Llama-3.1-70B-Instruct, Llama-3.1-405B-Instruct, Qwen2.5-72B-Instruct, Qwen3-235B-A22B-Instruct-2507, Qwen3-Coder-Next) were run on NVIDIA H200 GPUs, using 4 GPUs. One full evaluation pass per model takes approximately 3-4 GPU-hours and up to 10 hours for the thinking models.

E.1 ACCURACY RESULTS FOR PANDASSHIFT

The results are split by families and conditions (*Internal Only*, *External Repeat Internal*, *Conflicting External*, *Conflicting External + Instruct*). The Number of Samples per Documentation indicates the number of samples use cases for the operator in the SQL documentation in that particular run. For the *Conflicting External* and *Conflicting External + Instruct* conditions, values are averaged across different versions of the external knowledge. Direct Answer measures the accuracy of the model's direct answer against the gold direct answer. Orig Output measures the accuracy of running the Pandas code returned by the model against the gold output. New ([]) Output and New (AGG) Output similarly measure the accuracy of running the Pandas code returned by the model, assuming only the selection operator or aggregation operator was replaced, respectively. Orig Str measures the string-match accuracy of the Pandas code returned by the model against the gold Pandas code. New ([]) Str and New (AGG) Str similarly measure the string-match accuracy of the Pandas code returned by the model, assuming only the selection operator or aggregation operator was replaced, respectively. Lastly, Abs. Rate measures the model's abstention rate, i.e., the rate at which it returns NULL, indicating that it is unable to answer the question.

Table 15: Accuracy results for the Llama-3.1 family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Llama-3.1-8B-Instruct	*Internal Only*	–	52.88	98.08 ± 13.80	0.00 ± 0.00	0.00	0.38	0.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	54.81	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	54.42	45.58 ± 23.72	5.38 ± 8.91	16.35	5.58	45.58 ± 23.72	5.38 ± 8.91	16.35	5.58	0.00
	Conflicting External + Instruct	5	55.58	43.46 ± 25.15	6.54 ± 9.43	17.12	4.81	43.27 ± 25.10	6.54 ± 9.43	17.12	4.81	0.00
Llama-3.1-70B-Instruct	*Internal Only*	–	79.81	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	63.46	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	60.58	88.65 ± 19.06	0.58 ± 3.36	0.77	0.00	57.88 ± 29.88	0.58 ± 3.36	0.77	0.00	9.04
	Conflicting External + Instruct	5	56.54	54.62 ± 29.26	4.04 ± 8.54	14.23	0.00	42.50 ± 29.85	4.04 ± 8.54	14.23	0.00	19.81
Llama-3.1-405B-Instruct	*Internal Only*	–	90.38	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	74.04 ± 44.05	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	91.35	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	84.42	88.65 ± 13.87	4.62 ± 11.57	0.00	0.00	87.69 ± 13.74	4.62 ± 11.57	0.00	0.00	0.00
	Conflicting External + Instruct	5	85.96	47.12 ± 23.14	36.54 ± 27.37	3.46	0.00	47.12 ± 23.14	36.54 ± 27.37	3.46	0.00	0.00

Table 16: Accuracy results for the Qwen2.5 family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen2.5-7B-Instruct	*Internal Only*	–	55.77	94.23 ± 23.43	0.00 ± 0.00	0.00	0.38	81.73 ± 38.83	0.00 ± 0.00	0.00	0.38	0.00
	External Repeat Internal	5	50.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	50.00	0.00 ± 0.00	26.15 ± 9.28	56.15	0.00	0.00 ± 0.00	26.15 ± 9.28	56.15	0.00	0.96
	Conflicting External + Instruct	5	51.54	0.00 ± 0.00	22.50 ± 6.65	62.12	0.00	0.00 ± 0.00	22.50 ± 6.65	62.12	0.00	0.96
Qwen2.5-14B-Instruct	*Internal Only*	–	58.65	91.35 ± 28.25	0.00 ± 0.00	0.00	0.00	25.00 ± 43.51	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	60.58	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	0.00
	Conflicting External	5	58.85	0.00 ± 0.00	35.58 ± 16.48	49.81	0.00	0.00 ± 0.00	35.58 ± 16.48	49.81	0.00	0.00
	Conflicting External + Instruct	5	59.04	0.00 ± 0.00	35.96 ± 17.04	48.85	0.00	0.00 ± 0.00	35.96 ± 17.04	48.85	0.00	0.00
Qwen2.5-32B-Instruct	*Internal Only*	–	74.04	54.81 ± 50.01	0.00 ± 0.00	0.00	0.00	42.31 ± 49.64	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	69.23	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	68.08	2.88 ± 7.06	42.31 ± 24.18	37.88	0.00	2.31 ± 6.42	42.31 ± 24.18	37.88	0.00	0.38
	Conflicting External + Instruct	5	69.23	0.96 ± 4.30	41.54 ± 24.72	41.54	0.00	0.96 ± 4.30	41.54 ± 24.72	41.54	0.00	0.00
Qwen2.5-72B-Instruct	*Internal Only*	–	77.88	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	0.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	71.15	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	72.12	94.62 ± 8.91	0.00 ± 0.00	0.00	0.00	94.62 ± 8.91	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External + Instruct	5	72.31	90.00 ± 12.15	0.00 ± 0.00	0.38	0.00	89.42 ± 12.76	0.00 ± 0.00	0.38	0.00	0.00

Table 17: Accuracy results for the Qwen2.5-Coder family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen2.5-Coder-7B-Instruct	*Internal Only*	–	51.92	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	53.85	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	51.92	73.27 ± 26.38	0.00 ± 0.00	7.50	0.00	73.27 ± 26.38	0.00 ± 0.00	7.50	0.00	0.00
	Conflicting External + Instruct	5	52.12	69.81 ± 26.73	0.00 ± 0.00	9.62	0.00	69.81 ± 26.73	0.00 ± 0.00	9.62	0.00	0.00
Qwen2.5-Coder-14B-Instruct	*Internal Only*	–	59.62	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	87.50 ± 33.23	0.00 ± 0.00	0.00	0.38	0.00
	External Repeat Internal	5	52.88	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	99.04 ± 9.81	0.00 ± 0.00	0.00	0.38	0.00
	Conflicting External	5	54.42	1.35 ± 5.04	29.62 ± 10.79	51.54	0.00	1.35 ± 5.04	29.62 ± 10.79	51.54	0.00	0.00
	Conflicting External + Instruct	5	54.04	0.96 ± 4.30	29.81 ± 10.79	51.73	0.00	0.96 ± 4.30	29.81 ± 10.79	51.73	0.00	0.00
Qwen2.5-Coder-32B-Instruct	*Internal Only*	–	73.08	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	93.27 ± 25.18	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	67.31	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	68.85	79.23 ± 16.47	0.00 ± 0.00	0.96	0.00	79.23 ± 16.47	0.00 ± 0.00	0.96	0.00	0.00
	Conflicting External + Instruct	5	68.08	65.58 ± 24.21	0.00 ± 0.00	5.38	0.00	65.58 ± 24.21	0.00 ± 0.00	5.38	0.00	0.00

Table 18: Accuracy results for the Qwen3-Instruct family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-4B-Instruct-2507	*Internal Only*	–	57.69	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	51.92 ± 50.20	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	58.65	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	55.77	0.00 ± 0.00	11.15 ± 13.89	55.19	0.00	0.00 ± 0.00	11.15 ± 13.89	55.19	0.00	0.00
	Conflicting External + Instruct	5	54.42	0.00 ± 0.00	14.23 ± 15.93	54.62	0.00	0.00 ± 0.00	14.23 ± 15.93	54.62	0.00	0.00
Qwen3-30B-A3B-Instruct-2507	*Internal Only*	–	70.19	96.15 ± 19.32	0.00 ± 0.00	0.00	0.00	85.58 ± 35.30	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	70.19	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	71.35	1.73 ± 5.65	16.73 ± 12.80	77.12	0.00	1.73 ± 5.65	16.73 ± 12.80	77.12	0.00	0.00
	Conflicting External + Instruct	5	70.77	0.77 ± 3.86	19.42 ± 15.38	76.35	0.00	0.77 ± 3.86	19.42 ± 15.38	76.35	0.00	0.00
Qwen3-235B-A22B-Instruct-2507	*Internal Only*	–	87.50	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	25.96 ± 44.05	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	84.62	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	80.19	7.12 ± 13.05	59.42 ± 16.83	25.77	0.00	7.12 ± 13.05	59.42 ± 16.83	25.77	0.00	4.42
	Conflicting External + Instruct	5	73.85	0.00 ± 0.00	66.54 ± 18.37	20.38	0.00	0.00 ± 0.00	66.54 ± 18.37	20.38	0.00	12.50

Table 19: Accuracy results for the Qwen3-Thinking family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-4B-Thinking-2507	*Internal Only*	–	98.08	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	31.73 ± 46.77	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	96.15	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	96.35	30.77 ± 17.16	44.23 ± 22.76	4.04	0.00	29.23 ± 17.16	44.23 ± 22.76	4.04	0.00	0.19
	Conflicting External + Instruct	5	94.23	5.00 ± 9.14	69.62 ± 16.72	1.92	0.00	4.62 ± 8.47	69.62 ± 16.72	1.92	0.00	0.19
Qwen3-30B-A3B-Thinking-2507	*Internal Only*	–	96.15	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	5.77 ± 23.43	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	98.08	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	97.88	9.62 ± 10.79	24.23 ± 16.65	47.50	0.00	9.62 ± 10.79	24.23 ± 16.65	47.50	0.00	0.19
	Conflicting External + Instruct	5	96.92	4.62 ± 8.91	34.23 ± 17.77	44.23	0.00	4.62 ± 8.91	34.23 ± 17.77	44.23	0.00	0.00

Table 20: Accuracy results for the Qwen3-Coder family on PANDASSHIFT.

Model Name	Use Case	Number of Samples per Documentation	Direct Answer (%)	Orig Output (%)	New Output (%)	New ([]) Output (%)	New (AGG) Output (%)	Orig Str (%)	New Str (%)	New ([]) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Qwen3-Coder-30B-A3B-Instruct	*Internal Only*	–	68.27	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	48.08 ± 50.20	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	65.38	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	65.58	0.00 ± 0.00	17.88 ± 10.02	81.73	0.00	0.00 ± 0.00	17.88 ± 10.02	81.73	0.00	0.00
	Conflicting External + Instruct	5	66.73	0.00 ± 0.00	11.92 ± 9.86	85.00	0.00	0.00 ± 0.00	11.92 ± 9.86	85.00	0.00	0.00
Qwen3-Coder-Next	*Internal Only*	–	81.73	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	94.23 ± 23.43	0.00 ± 0.00	0.00	0.00	0.00
	External Repeat Internal	5	80.77	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Conflicting External	5	79.81	18.46 ± 17.34	11.15 ± 11.09	46.92	0.00	18.46 ± 17.34	11.15 ± 11.09	46.92	0.00	0.00
	Conflicting External + Instruct	5	79.81	10.00 ± 15.77	12.31 ± 11.25	53.08	0.00	9.81 ± 15.76	12.31 ± 11.25	53.08	0.00	0.00

E.2 KNOWLEDGE ANCHOR FOR PANDASSHIFT

Table 21: Knowledge anchor under PANDASSHIFT. Values are bounded between $[-2, 2]$ where -2 is full leakage and 2 is no leakage at all. Cell shading encodes value on a fixed scale from red (-2, max leakage) to blue ($+2$, no leakage).

Model Name	*Internal Only*	*External Repeat Internal*	*Conflicting External*	*Conflicting External + Instruct*
Llama-3.1-8B-Instruct	-1.9925 ± 0.0941	-1.9988 ± 0.0113	-0.6404 ± 0.8602	-0.5520 ± 0.9150
Llama-3.1-70B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.7325 ± 0.4027	-1.0432 ± 0.9856
Llama-3.1-405B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.5907 ± 0.5044	-0.1911 ± 1.0022
Qwen2.5-7B-Instruct	-1.7580 ± 0.9084	-2.0000 ± 0.0000	0.6567 ± 1.3619	0.5547 ± 1.4606
Qwen2.5-14B-Instruct	-1.9807 ± 0.2780	-1.9840 ± 0.2301	0.8187 ± 1.2380	0.8292 ± 1.2388
Qwen2.5-32B-Instruct	-1.8593 ± 0.7388	-2.0000 ± 0.0000	0.9619 ± 1.1251	0.9431 ± 1.1706
Qwen2.5-72B-Instruct	-1.9808 ± 0.2774	-1.9980 ± 0.0285	-1.6961 ± 0.3966	-1.4509 ± 0.6441
Qwen2.5-Coder-7B-Instruct	-1.9807 ± 0.2773	-2.0000 ± 0.0001	-0.5416 ± 1.4808	-0.5307 ± 1.4951
Qwen2.5-Coder-14B-Instruct	-1.9808 ± 0.2762	-1.9826 ± 0.2499	0.7456 ± 1.2210	0.7656 ± 1.2164
Qwen2.5-Coder-32B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.4432 ± 0.8351	-1.1717 ± 1.1071
Qwen3-4B-Instruct-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0000	0.3955 ± 1.7332	0.4853 ± 1.6695
Qwen3-30B-A3B-Instruct-2507	-1.9808 ± 0.2774	-2.0000 ± 0.0000	0.3128 ± 1.6618	0.3771 ± 1.6501
Qwen3-235B-A22B-Instruct-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0000	1.1208 ± 0.7656	1.5404 ± 0.5861
Qwen3-4B-Thinking-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0001	0.5145 ± 0.8495	1.7404 ± 0.4415
Qwen3-30B-A3B-Thinking-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0000	0.4039 ± 1.3671	0.7556 ± 1.2432
Qwen3-Coder-30B-A3B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	0.3637 ± 1.6644	0.2667 ± 1.7595
Qwen3-Coder-Next	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-0.0502 ± 1.3894	0.1485 ± 1.4853

E.3 KNOWLEDGE SHIFT FOR PANDASSHIFT

Table 22: Knowledge shift under PANDASSHIFT. *External Repeat Internal* compares the probability of a token when it belongs to internal knowledge against *Internal Only*; *Conflicting External* and *Conflicting External + Instruct* compare the probability of a token when it belongs to external knowledge against *Internal Only*.

Model Name	*External Repeat Internal*	*Conflicting External*	*Conflicting External + Instruct*
Llama-3.1-8B-Instruct	1.0191 ± 0.0427	1.2979 ± 0.1910	1.3154 ± 0.2014
Llama-3.1-70B-Instruct	1.0003 ± 0.0011	1.0642 ± 0.0973	1.2219 ± 0.2273
Llama-3.1-405B-Instruct	1.0000 ± 0.0001	1.1003 ± 0.1248	1.4391 ± 0.2541
Qwen2.5-7B-Instruct	1.0912 ± 0.2693	1.6261 ± 0.3689	1.6025 ± 0.3908
Qwen2.5-14B-Instruct	1.0026 ± 0.0767	1.6827 ± 0.3319	1.6867 ± 0.3293
Qwen2.5-32B-Instruct	1.0553 ± 0.1981	1.7049 ± 0.2626	1.7087 ± 0.2778
Qwen2.5-72B-Instruct	1.0038 ± 0.0539	1.0714 ± 0.0973	1.1269 ± 0.1539
Qwen2.5-Coder-7B-Instruct	1.0070 ± 0.0694	1.3508 ± 0.3595	1.3537 ± 0.3634
Qwen2.5-Coder-14B-Instruct	1.0007 ± 0.0030	1.6656 ± 0.3033	1.6687 ± 0.3007
Qwen2.5-Coder-32B-Instruct	1.0015 ± 0.0027	1.1306 ± 0.2000	1.1890 ± 0.2631
Qwen3-4B-Instruct-2507	1.0005 ± 0.0023	1.5320 ± 0.3773	1.5513 ± 0.3605
Qwen3-30B-A3B-Instruct-2507	1.0048 ± 0.0693	1.5767 ± 0.4169	1.5927 ± 0.4141
Qwen3-235B-A22B-Instruct-2507	1.0000 ± 0.0000	1.7797 ± 0.1911	1.8850 ± 0.1465
Qwen3-4B-Thinking-2507	1.0000 ± 0.0000	1.6116 ± 0.2087	1.9031 ± 0.1265
Qwen3-30B-A3B-Thinking-2507	1.0000 ± 0.0000	1.5896 ± 0.3338	1.6789 ± 0.3043
Qwen3-Coder-30B-A3B-Instruct	1.0000 ± 0.0000	1.5902 ± 0.4166	1.5653 ± 0.4408
Qwen3-Coder-Next	1.0004 ± 0.0007	1.4544 ± 0.3161	1.4986 ± 0.3370

E.4 Original Operator Token Distribution for PandasShift

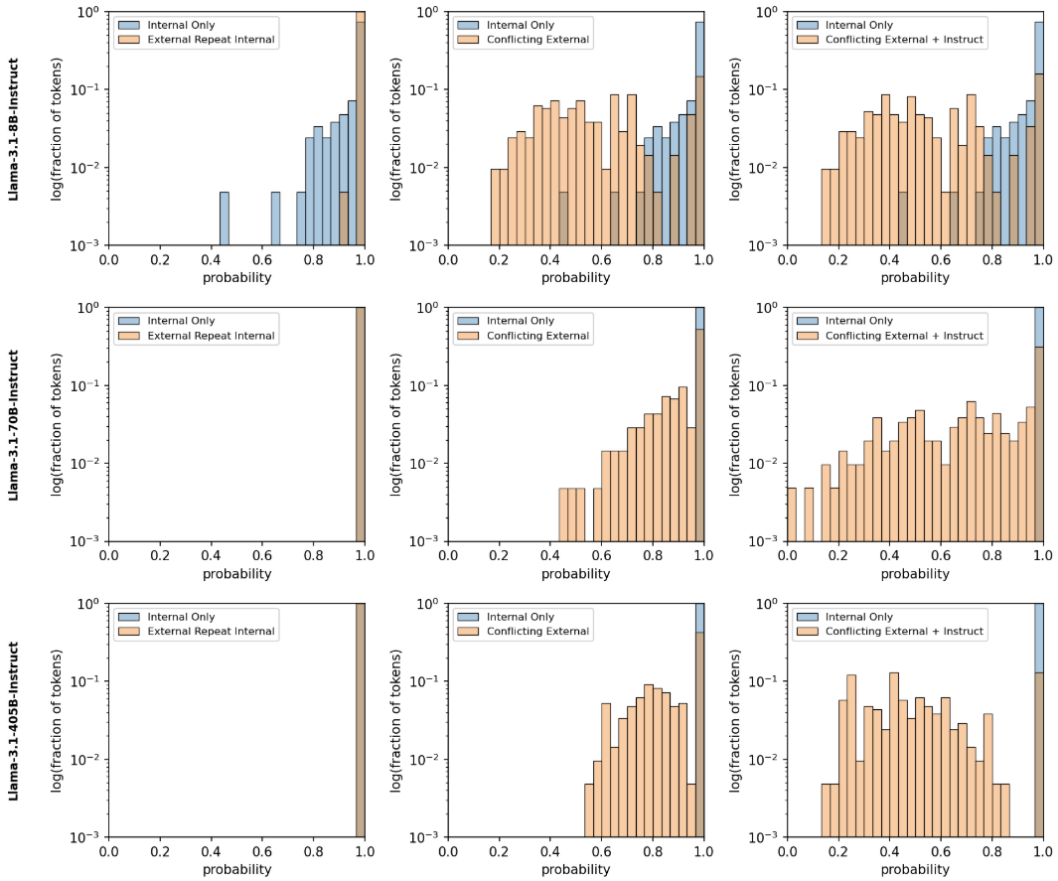

Figure 24: Distribution of the per-token probability mass placed on the *original* version operator for Llama3.1 family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

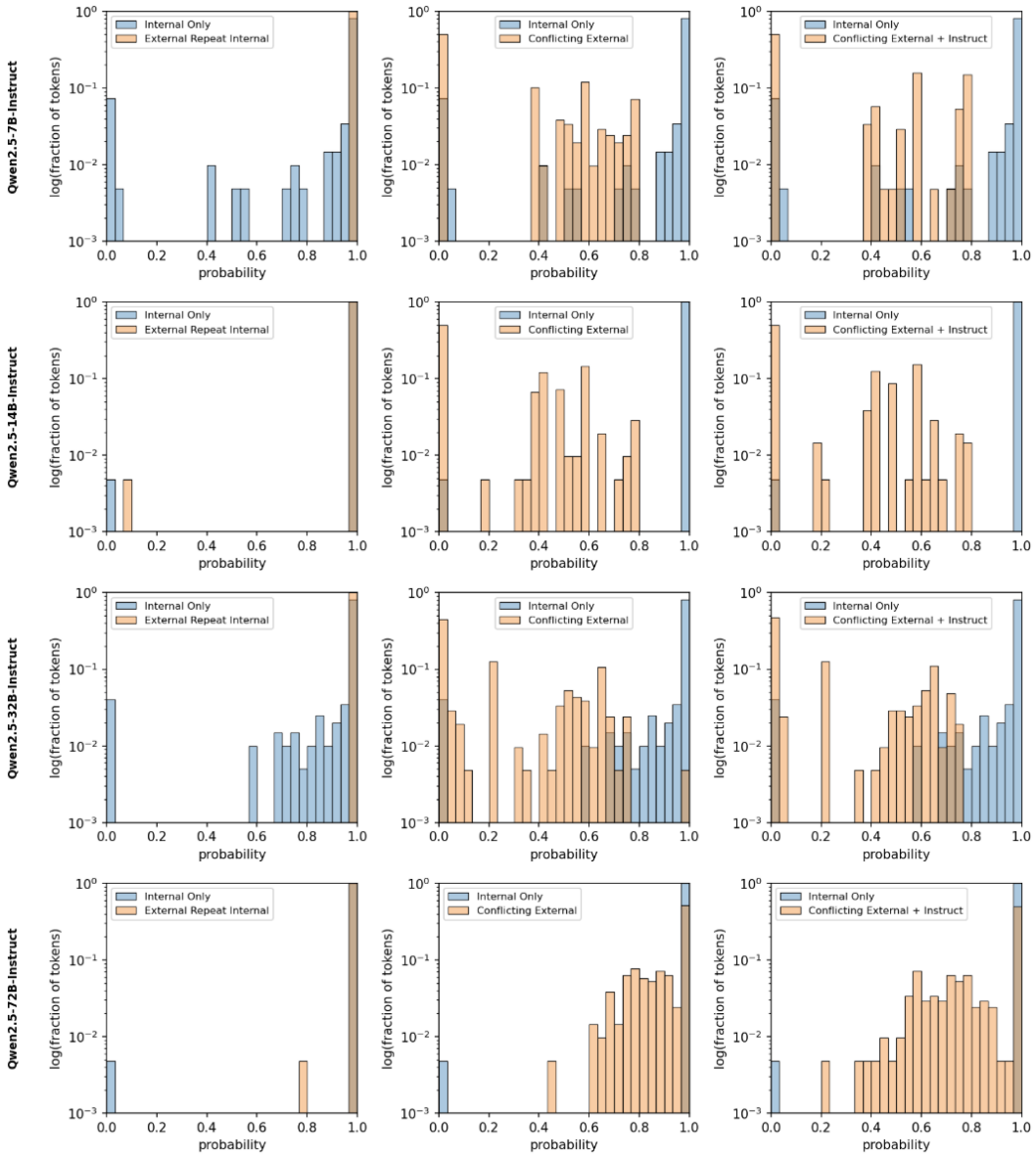

Figure 25: Distribution of the per-token probability mass placed on the *original* version operator for Qwen2.5-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

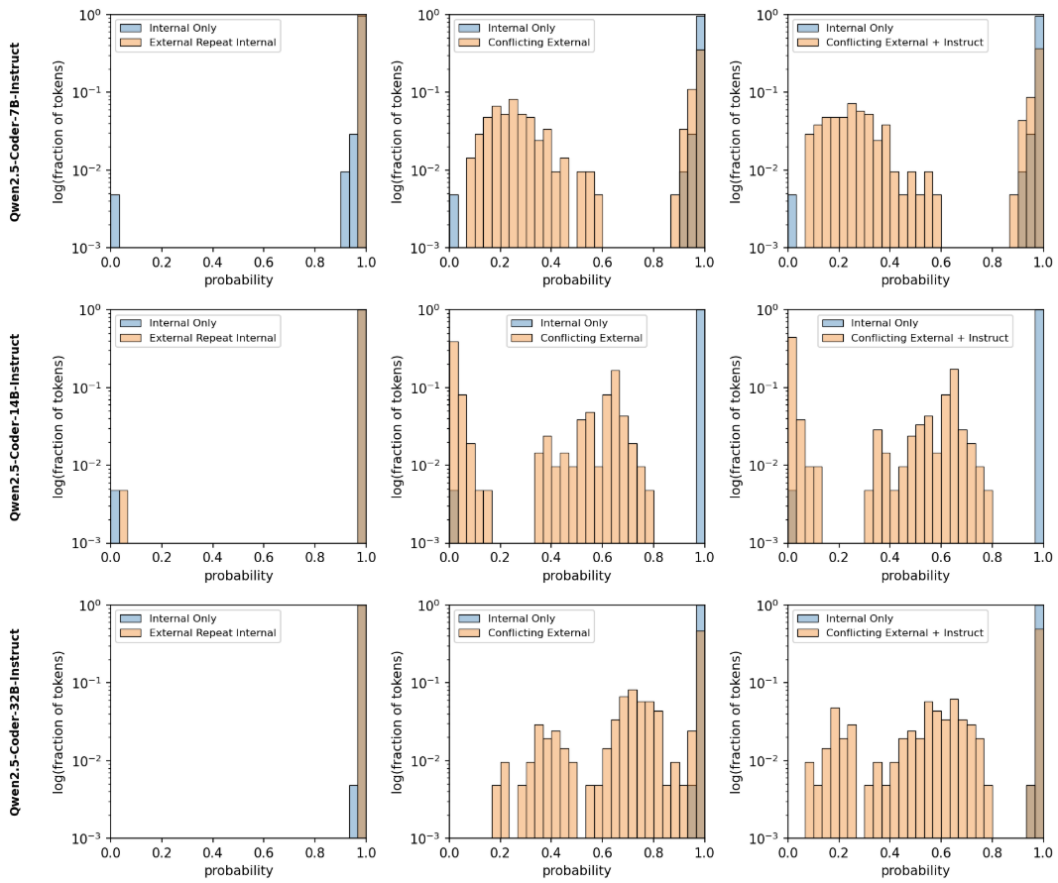

Figure 26: Distribution of the per-token probability mass placed on the *original* version operator for Qwen2.5-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

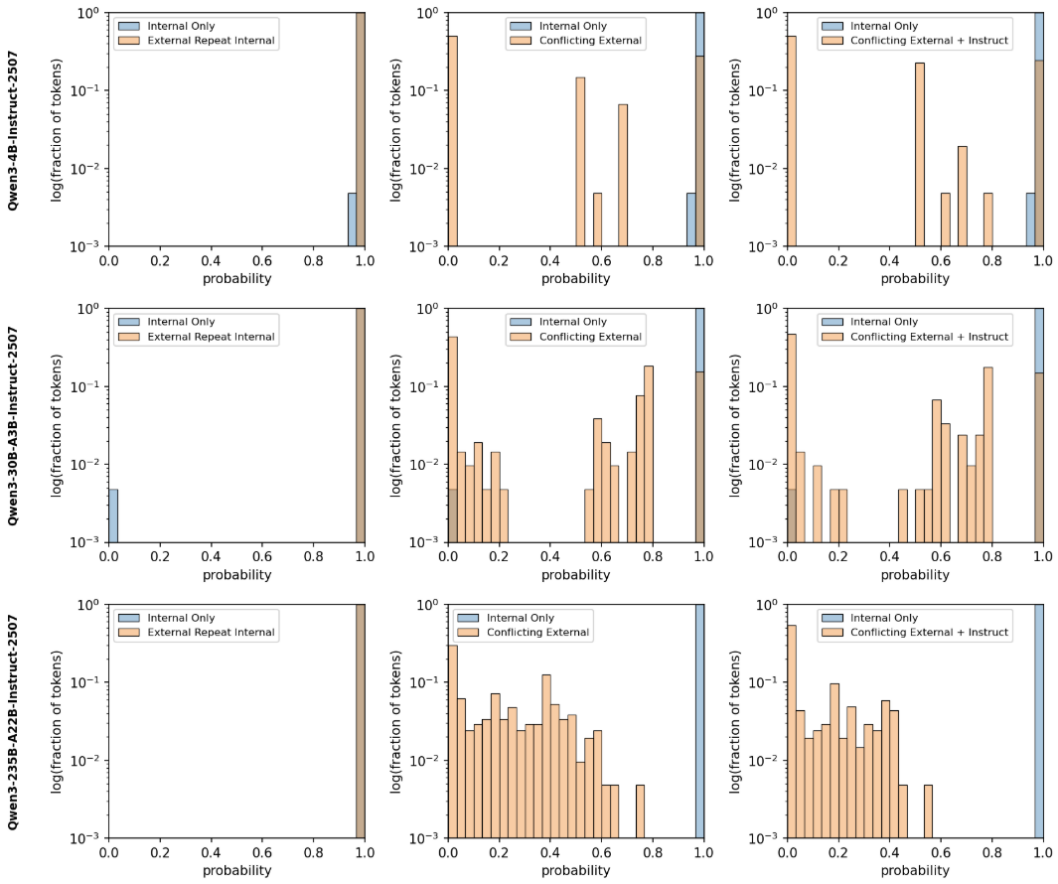

Figure 27: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

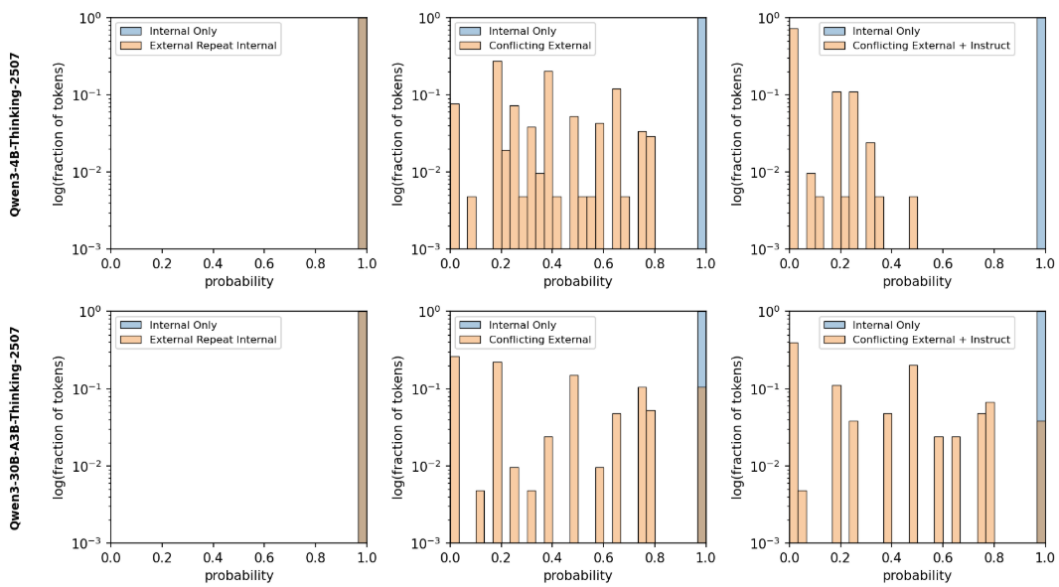

Figure 28: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Thinking family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

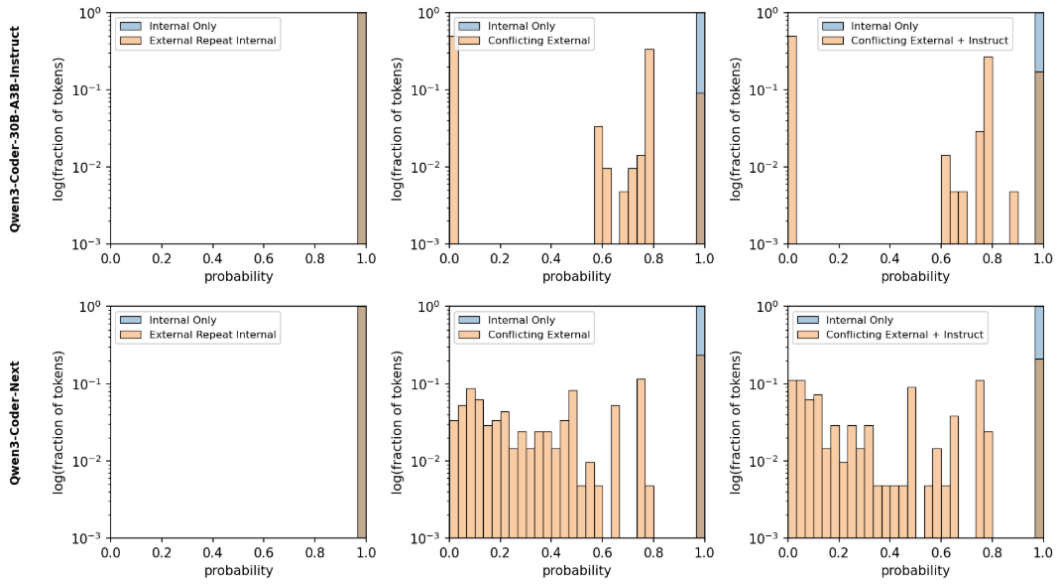

Figure 29: Distribution of the per-token probability mass placed on the *original* version operator for Qwen3-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

E.5 New Operator Token Distribution for PandasShift

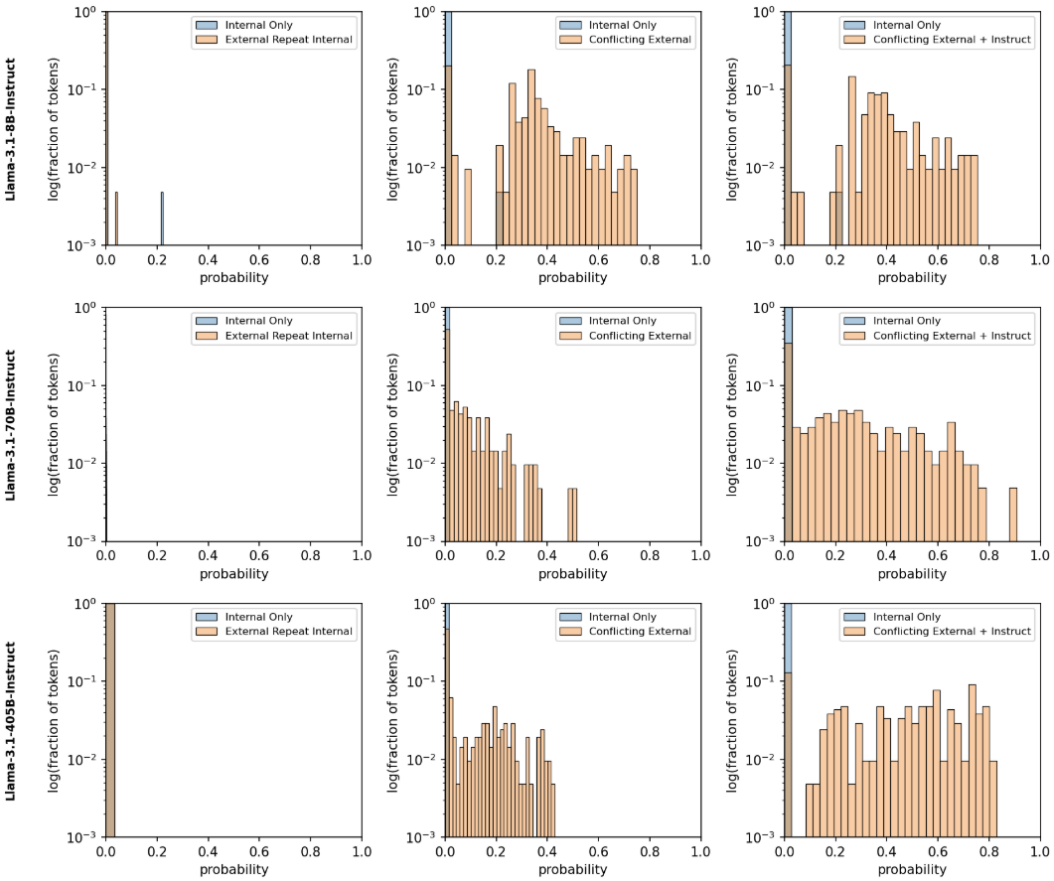

Figure 30: Distribution of the per-token probability mass placed on the *new* version operator for Llama3.1 family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

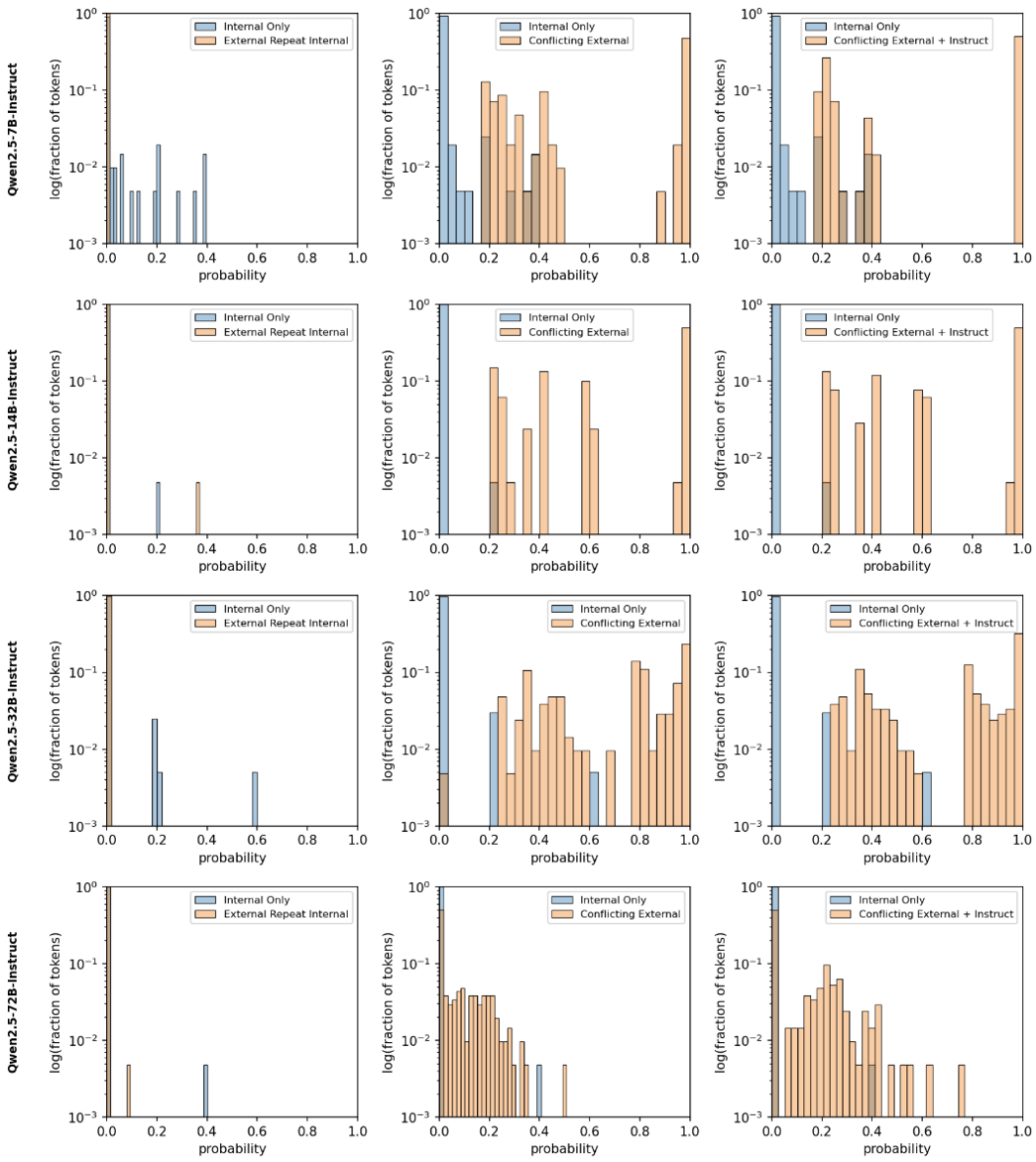

Figure 31: Distribution of the per-token probability mass placed on the *new* version operator for Qwen2.5-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

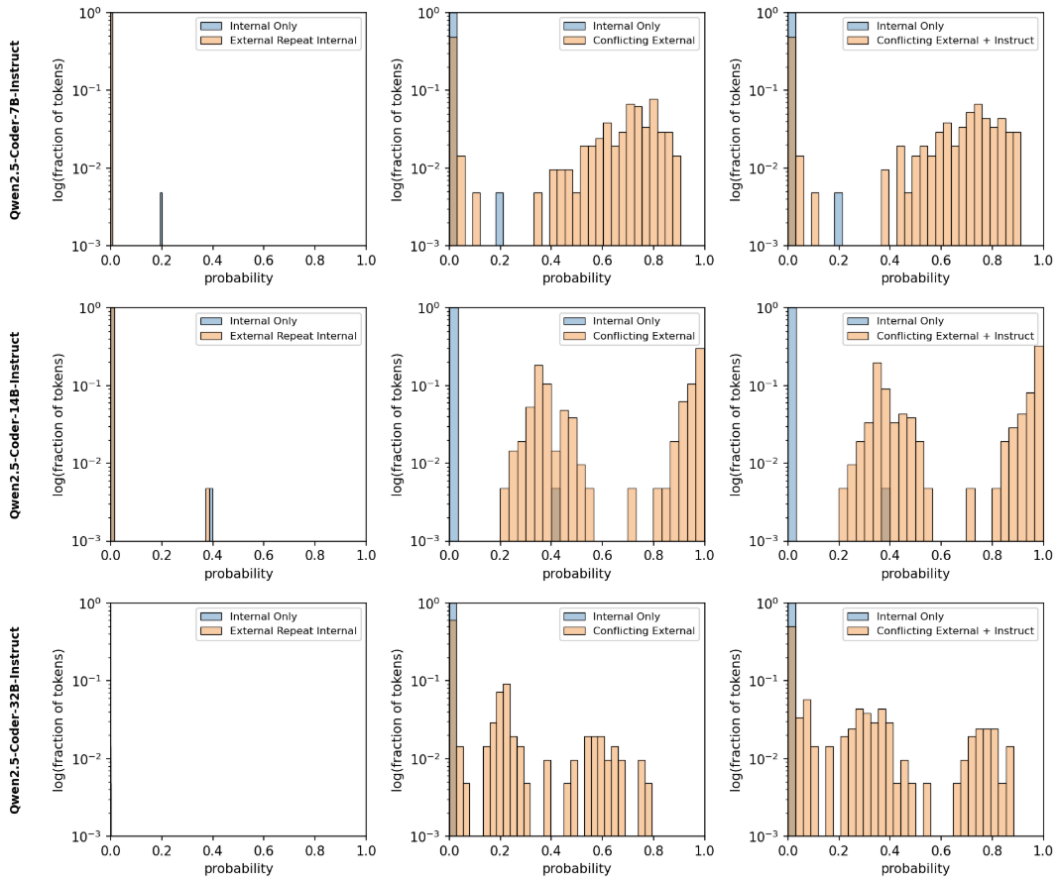

Figure 32: Distribution of the per-token probability mass placed on the *new* version operator for Qwen2.5-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

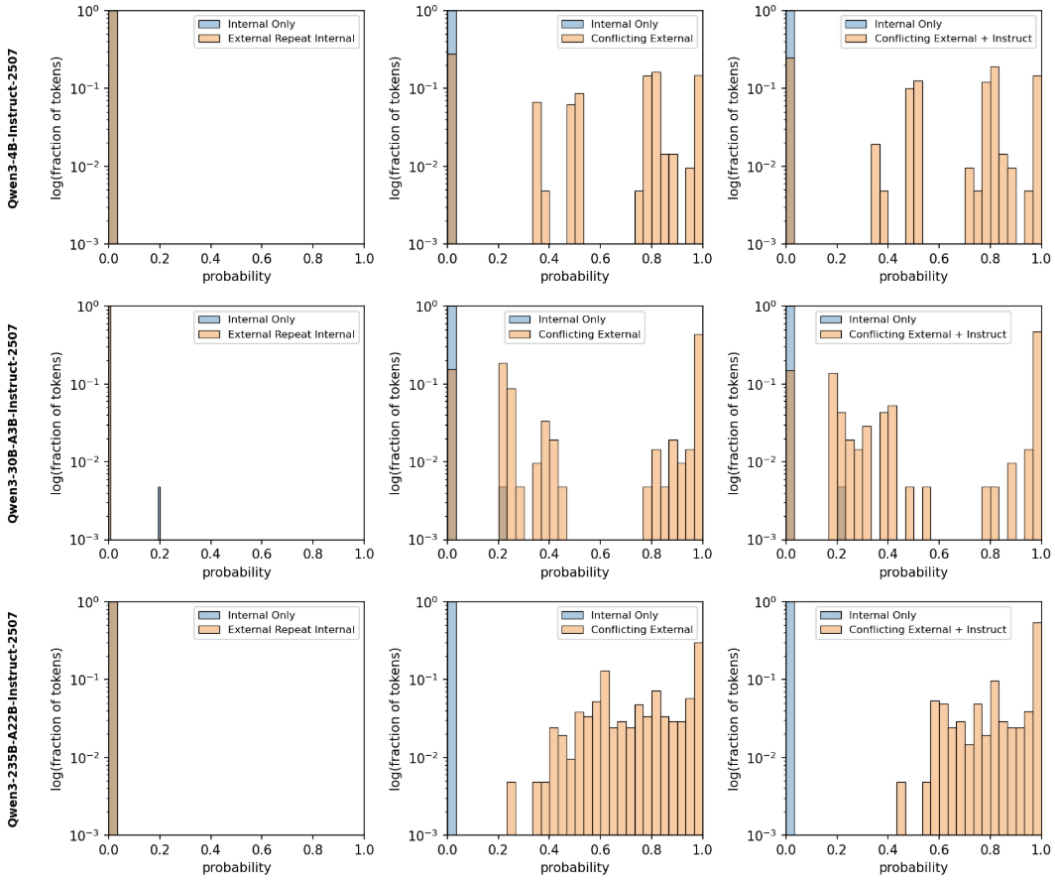

Figure 33: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Instruct family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

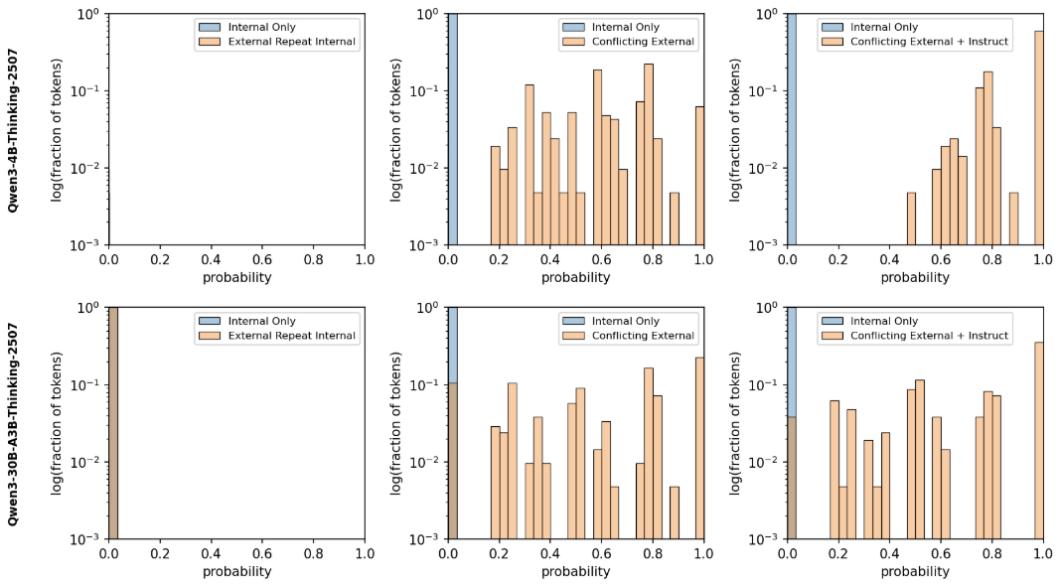

Figure 34: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Thinking family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

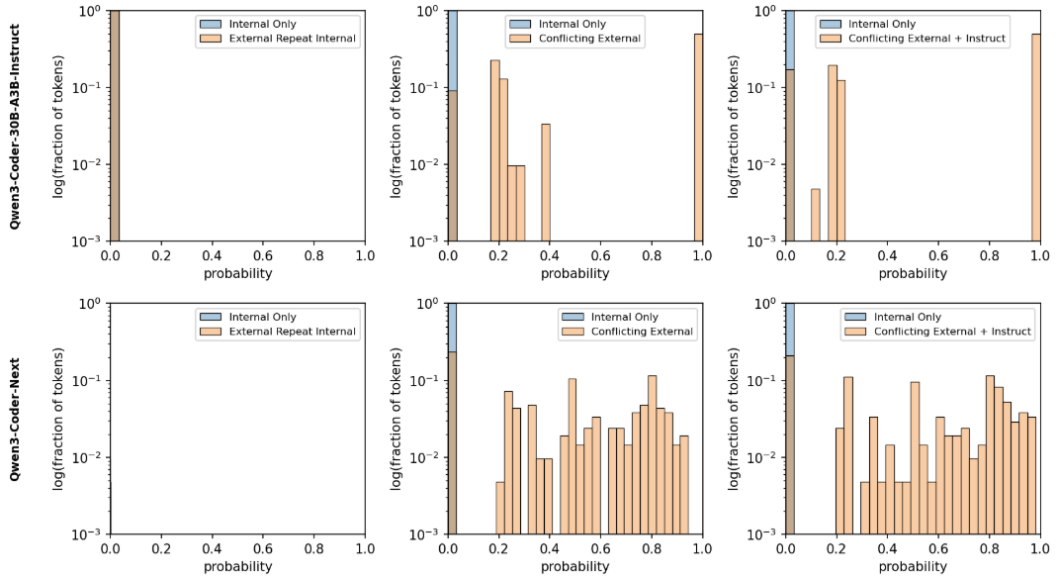

Figure 35: Distribution of the per-token probability mass placed on the *new* version operator for Qwen3-Coder family. The y-axis is log(fraction of tokens); the x-axis is the raw probability value of each operator token; the *Internal Only* distribution is blue, the other condition orange.

F FULL UNLEARNING RESULTS

A100 80GB node, taking about 3 GPU-hours per configuration.
Table 23 shows the accuracy results after running gradient-ascent unlearning on SQLSHIFT. The row highlighted in yellow indicates the model with the highest success in unlearning. In most models, unlearning was not successful in removing the relevant SQL knowledge, as indicated by the high accuracy rate under the *Internal Only* condition; while the model performs poorly under string match (Orig Str), it maintains good performance when the query is executed using SQLite (Orig Output), because SQLite is case-insensitive while our string match is case-sensitive. This shows that the model successfully unlearned the format where SQL operators are presented in pure upper-case (the syntax in the SQL documentation), but not the case where the SQL operators could be in a mix of lower and upper case. Coder models unlearned SQL knowledge less than non-Coder models. The rows highlighted in cyan indicate the case of highest leakage; after unlearning, the model seems to default back to internal knowledge even more than before, corroborated by the Knowledge Anchor and Knowledge Shift metrics in Tables 24 and 25.

Table 23: Accuracy results after gradient-ascent unlearning of the SQL question-answering data. Where applicable we only use 5 sample cases in each operator documentation; for *Conflicting External* and *Conflicting External + Instruct* values are averaged across the five versions of new knowledge. **Orig Output / New Output** measure the accuracy of running the returned SQL query against the gold output (with new knowledge substituted in for **New Output**).

Condition	Model Name	Direct Answer (%)	Orig Output (%)	New Output (%)	Orig Str (%)	New Str (%)	New (SEL) Str (%)	New (AGG) Str (%)	Abs. Rate (%)
Internal Only	**Qwen2.5-32B-Instruct**	66.35	0.96 ± 9.81	0.00 ± 0.00	0.96 ± 9.81	0.00 ± 0.00	0.00	0.00	1.92
	Qwen2.5-Coder-32B-Instruct	74.04	99.04 ± 9.81	0.00 ± 0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Instruct-2507	69.23	99.04 ± 9.81	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Thinking-2507	94.23	98.08 ± 13.80	0.00 ± 0.00	97.12 ± 16.82	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-Coder-30B-A3B-Instruct	62.50	100.00 ± 0.00	0.00 ± 0.00	76.92 ± 42.34	0.00 ± 0.00	0.00	0.00	0.00
External Repeat Internal doclen 5	**Qwen2.5-32B-Instruct**	59.62	58.65 ± 49.48	0.00 ± 0.00	58.65 ± 49.48	0.00 ± 0.00	0.00	0.00	0.00
	Qwen2.5-Coder-32B-Instruct	65.38	100.00 ± 0.00	0.00 ± 0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Instruct-2507	65.38	99.04 ± 9.81	0.00 ± 0.00	62.50 ± 48.65	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Thinking-2507	94.23	99.04 ± 9.81	0.00 ± 0.00	99.04 ± 9.81	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-Coder-30B-A3B-Instruct	60.58	100.00 ± 0.00	0.00 ± 0.00	100.00 ± 0.00	0.00 ± 0.00	0.00	0.00	0.00
Conflicting External doclen 5	**Qwen2.5-32B-Instruct**	62.69	80.96 ± 25.60	0.00 ± 0.00	80.58 ± 25.61	0.00 ± 0.00	3.46	0.00	0.58
	Qwen2.5-Coder-32B-Instruct	64.42	98.65 ± 5.76	0.00 ± 0.00	98.65 ± 5.76	0.00 ± 0.00	0.00	0.38	0.00
	Qwen3-30B-A3B-Instruct-2507	65.58	99.04 ± 9.81	0.00 ± 0.00	26.73 ± 29.97	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Thinking-2507	93.65	86.35 ± 20.72	0.00 ± 0.00	86.35 ± 20.72	0.00 ± 0.00	3.85	0.00	0.00
	Qwen3-Coder-30B-A3B-Instruct	60.19	94.81 ± 8.81	4.23 ± 8.21	94.81 ± 8.81	4.23 ± 8.21	0.19	0.00	0.00
Conflicting External + Instruct doclen 5	**Qwen2.5-32B-Instruct**	60.19	80.19 ± 26.07	0.00 ± 0.00	80.19 ± 26.07	0.00 ± 0.00	2.69	0.00	4.42
	Qwen2.5-Coder-32B-Instruct	63.08	90.58 ± 10.41	0.00 ± 0.00	90.58 ± 10.41	0.00 ± 0.00	1.73	0.19	0.00
	Qwen3-30B-A3B-Instruct-2507	65.00	99.23 ± 7.84	0.00 ± 0.00	55.58 ± 43.44	0.00 ± 0.00	0.00	0.00	0.00
	Qwen3-30B-A3B-Thinking-2507	93.65	53.46 ± 23.88	0.58 ± 3.36	53.46 ± 23.88	0.58 ± 3.36	22.69	0.00	0.00
	Qwen3-Coder-30B-A3B-Instruct	61.35	85.77 ± 20.03	4.23 ± 8.21	85.77 ± 20.03	4.23 ± 8.21	6.73	0.38	0.00

Table 24: Knowledge Anchor after unlearning. Cell shading encodes the extent of leakage, red (−2, max leakage) to blue (+2, no leakage).

Model Name	*Internal Only*	*External Repeat Internal*	*Conflicting External*	*Conflicting External + Instruct*
Qwen2.5-32B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.6094 ± 0.7085	-1.6664 ± 0.6701
Qwen2.5-Coder-32B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.9519 ± 0.1465	-1.8083 ± 0.2574
Qwen3-30B-A3B-Instruct-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-2.0000 ± 0.0000
Qwen3-30B-A3B-Thinking-2507	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.7774 ± 0.5625	-1.0996 ± 1.0826
Qwen3-Coder-30B-A3B-Instruct	-2.0000 ± 0.0000	-2.0000 ± 0.0000	-1.8218 ± 0.3083	-1.5782 ± 0.6042

Table 25: Knowledge Shift after unlearning.

Model Name	*External Repeat Internal*	*Conflicting External*	*Conflicting External + Instruct*
Qwen2.5-32B-Instruct	1.0069 ± 0.0288	1.0069 ± 0.0165	1.0053 ± 0.0139
Qwen2.5-Coder-32B-Instruct	0.9992 ± 0.0121	1.0116 ± 0.0329	1.0463 ± 0.0624
Qwen3-30B-A3B-Instruct-2507	1.0000 ± 0.0003	1.0000 ± 0.0000	1.0000 ± 0.0000
Qwen3-30B-A3B-Thinking-2507	1.0000 ± 0.0000	1.0540 ± 0.1368	1.2136 ± 0.2555
Qwen3-Coder-30B-A3B-Instruct	1.0000 ± 0.0000	1.0442 ± 0.0766	1.1023 ± 0.1431

G VINTAGE-LLM (TALKIE)

This appendix gives the two-phase knowledge-injection procedure, the full Talkie training results (no further training, Phase 1 continued pre-training, Phase 2 LoRA supervised fine-tuning, and the full pipeline), and the Talkie Knowledge Anchor tables for SQLSHIFT.

G.1 TWO-PHASE KNOWLEDGE INJECTION

Phase 1 is the initial SQL knowledge injection: continued pre-training of Talkie on the SQL operator documentation. Each document is wrapped in Talkie's chat template before encoding, and cross-entropy loss is computed on the next-token prediction. We train with a learning rate of 1×10^{-5} using AdamW, varying the number of epochs (5, 10) and the number of sample use cases per documentation (1, 5, 10). *Phase 2* performs LoRA supervised fine-tuning using a training set adapted from WIKISQL. LoRA modules are inserted into the query, key, value, and output projections of every attention block, while base weights remain frozen (HuggingFace PEFT HuggingFace (2026)). Each input consists of the instruction prompt, target table, and natural-language question, following the Baseline 1 prompt format; the output target is a single structured string `{ANSWER: <ground truth answer>, SQL: <ground truth query>}`. Loss is computed only over the output target tokens (prompt labels masked to -100). We train with a learning rate of 1×10^{-5} using AdamW, varying epochs (5, 10) and the number of training examples (50, 100, 200). Talkie fine-tuning (Phase 1 CPT, Phase 2 LoRA SFT) were run on a single A100 80GB node, taking about 3 GPU-hours per configuration.

G.2 TALKIE TRAINING RESULTS

Table 26: No Further Training at All.

Phase 1 Num Epochs	Phase 2 Num Epochs	Phase 1 Num Samples	Phase 2 Num Train	Direct Answer (%)	Output (%)	Abs. Rate (%)
–	–	–	–	0.00	0.00 ± 0.00	100.00

Table 27: Phase 1 Continued Pre-Training with SQL Documentations.

Phase 1 Num Epochs	Phase 2 Num Epochs	Phase 1 Num Samples	Phase 2 Num Train	Direct Answer (%)	Output (%)	Abs. Rate (%)
5	–	1	–	0.00	0.00 ± 0.00	100.00
5	–	5	–	0.00	0.00 ± 0.00	100.00
5	–	10	–	0.00	0.00 ± 0.00	100.00
10	–	1	–	0.00	0.00 ± 0.00	100.00
10	–	5	–	0.00	0.00 ± 0.00	100.00
10	–	10	–	0.00	0.00 ± 0.00	100.00

Table 28: Phase 2 LoRA Supervised Fine-Tuning using QA data.

Phase 1 Num Epochs	Phase 2 Num Epochs	Phase 1 Num Samples	Phase 2 Num Train	Direct Answer (%)	Output (%)	Abs. Rate (%)
–	5	–	50	10.58	70.19 ± 45.96	5.77
–	5	–	100	10.58	88.46 ± 32.10	16.35
–	5	–	200	13.46	73.08 ± 44.57	1.92
–	10	–	50	9.62	76.92 ± 42.34	12.50
–	10	–	100	12.50	89.42 ± 30.90	17.31
–	10	–	200	20.19	89.42 ± 30.90	2.88

Table 29: Full pipeline (Phase 1 + Phase 2).

Phase 1 Num Epochs	Phase 2 Num Epochs	Phase 1 Num Samples	Phase 2 Num Train	Direct Answer (%)	Output (%)	Abs. Rate (%)
5	5	1	50	9.62	39.42 ± 49.11	10.58
5	5	1	100	10.58	65.38 ± 47.80	2.88
5	5	1	200	22.12	88.46 ± 32.10	0.96
5	5	5	50	11.54	76.92 ± 42.34	8.65
5	5	5	100	13.46	77.88 ± 41.70	6.73
5	5	5	200	15.38	45.19 ± 50.01	6.73
5	5	10	50	6.73	65.38 ± 47.80	2.88
5	5	10	100	10.58	72.12 ± 45.06	1.92
5	5	10	200	17.31	75.96 ± 42.94	4.81
5	10	1	50	9.62	44.23 ± 49.91	2.88
5	10	1	100	8.65	47.12 ± 50.16	13.46
5	10	1	200	14.42	61.54 ± 48.89	6.73
5	10	5	50	11.54	38.46 ± 48.89	3.85
5	10	5	100	14.42	94.23 ± 23.43	1.92
5	10	5	200	20.19	96.15 ± 19.32	0.96
5	10	10	50	8.65	54.81 ± 50.01	2.88
5	10	10	100	12.50	59.62 ± 49.30	7.69
5	10	10	200	19.23	68.27 ± 46.77	2.88
10	5	1	50	9.62	81.73 ± 38.83	0.96
10	5	1	100	9.62	73.08 ± 44.57	7.69
10	5	1	200	20.19	68.27 ± 46.77	0.00
10	5	5	50	8.65	79.81 ± 40.34	1.92
10	5	5	100	10.58	81.73 ± 38.83	10.58
10	5	5	200	15.38	90.38 ± 29.62	5.77
10	5	10	50	9.62	66.35 ± 47.48	2.88
10	5	10	100	16.35	78.85 ± 41.04	10.58
10	5	10	200	17.31	49.04 ± 50.23	2.88
10	10	1	50	9.62	67.31 ± 47.14	3.85
10	10	1	100	13.46	89.42 ± 30.90	6.73
10	10	1	200	21.15	85.58 ± 35.30	7.69
10	10	5	50	11.54	47.12 ± 50.16	3.85
10	10	5	100	8.65	68.27 ± 46.77	7.69
10	10	5	200	15.38	80.77 ± 39.60	12.50
10	10	10	50	6.73	53.85 ± 50.09	2.88
10	10	10	100	12.50	78.85 ± 41.04	2.88
10	10	10	200	14.42	68.27 ± 46.77	1.92

G.3 TALKIE KNOWLEDGE ANCHOR FOR SQLSHIFT

Table 30: Phase 2 LoRA Supervised Fine-Tuning using QA data.

Phase 1 Num Epochs	Phase 2 Num Epochs	Phase 1 Num Samples	Phase 2 Num Train	*Internal Only*	*External Repeat Internal*	*Conflicting External*	*Conflicting External + Instruct*
–	5	–	50	-2 ± 0.0000	-1.9895 ± 0.1392	-1.973 ± 0.1210	-1.9484 ± 0.1961
–	5	–	100	-2 ± 0.0000	-2 ± 0.0000	-1.9499 ± 0.1521	-1.8945 ± 0.5214
–	5	–	200	-2 ± 0.0000	-2 ± 0.0000	-1.9935 ± 0.0513	-1.8929 ± 0.3281
–	10	–	50	-2 ± 0.0000	-2 ± 0.0000	-1.9991 ± 0.0097	-1.993 ± 0.0348
–	10	–	100	-2 ± 0.0000	-2 ± 0.0000	-1.9845 ± 0.0707	-1.9522 ± 0.3176
–	10	–	200	-2 ± 0.0000	-1.999 ± 0.0085	-1.9771 ± 0.0919	-1.9782 ± 0.1130

Table 31: Full pipeline (Phase 1 + Phase 2).

Phase 1 Num Epochs	**Phase 2 Num Epochs**	**Phase 1 Num Samples**	**Phase 2 Num Train**	***Internal Only***	***External Repeat Internal***	***Conflicting External***	***Conflicting External + Instruct***
5	5	1	50	-2 ± 0.0000	-2 ± 0.0000	-1.9506 ± 0.2197	-1.9463 ± 0.2517
5	5	1	100	-2 ± 0.0000	-2 ± 0.0000	-1.956 ± 0.2261	-1.636 ± 0.9587
5	5	1	200	-2 ± 0.0000	-2 ± 0.0000	-1.8008 ± 0.6876	-1.8438 ± 0.6932
5	5	5	50	-2 ± 0.0000	-2 ± 0.0003	-1.9418 ± 0.2295	-1.9558 ± 0.1861
5	5	5	100	-2 ± 0.0000	-1.9992 ± 0.0060	-1.91 ± 0.3446	-1.8472 ± 0.5008
5	5	5	200	-2 ± 0.0000	-2 ± 0.0000	-1.8113 ± 0.2570	-1.6595 ± 1.0245
5	5	10	50	-2 ± 0.0000	-2 ± 0.0000	-1.8843 ± 0.3324	-1.982 ± 0.1017
5	5	10	100	-2 ± 0.0000	-1.9959 ± 0.0349	-1.8956 ± 0.5109	-1.6963 ± 0.7870
5	5	10	200	-2 ± 0.0000	-2 ± 0.0000	-1.7874 ± 0.5726	-1.967 ± 0.1303
5	10	1	50	-2 ± 0.0000	-2 ± 0.0000	-1.9322 ± 0.2287	-1.9376 ± 0.3415
5	10	1	100	-2 ± 0.0000	-2 ± 0.0000	-1.9673 ± 0.1207	-1.9363 ± 0.2036
5	10	1	200	-2 ± 0.0000	-1.9979 ± 0.0150	-1.9501 ± 0.1971	-1.9871 ± 0.0978
5	10	5	50	-2 ± 0.0000	-1.9966 ± 0.0266	-1.9643 ± 0.1773	-1.943 ± 0.2548
5	10	5	100	-2 ± 0.0000	-2 ± 0.0000	-1.916 ± 0.4020	-1.95 ± 0.3064
5	10	5	200	-2 ± 0.0000	-2 ± 0.0000	-1.8052 ± 0.6037	-1.8913 ± 0.3206
5	10	10	50	-2 ± 0.0000	-2 ± 0.0000	-1.9689 ± 0.1399	-1.9716 ± 0.1195
5	10	10	100	-2 ± 0.0000	-2 ± 0.0000	-1.9309 ± 0.1507	-1.9259 ± 0.2445
5	10	10	200	-2 ± 0.0000	-2 ± 0.0000	-1.9679 ± 0.1578	-1.9163 ± 0.3076
10	5	1	50	-2 ± 0.0000	-2 ± 0.0000	-1.9459 ± 0.2493	-1.8462 ± 0.3508
10	5	1	100	-2 ± 0.0000	-2 ± 0.0004	-1.9472 ± 0.3123	-1.9046 ± 0.3535
10	5	1	200	-2 ± 0.0000	-2 ± 0.0000	-1.8755 ± 0.4690	-1.8452 ± 0.3108
10	5	5	50	-2 ± 0.0000	-1.9987 ± 0.0159	-1.9682 ± 0.1458	-1.9772 ± 0.0988
10	5	5	100	-2 ± 0.0000	-2 ± 0.0000	-1.9027 ± 0.3410	-1.9259 ± 0.2290
10	5	5	200	-2 ± 0.0000	-1.9821 ± 0.1036	-1.7445 ± 0.6427	-1.9451 ± 0.2588
10	5	10	50	-2 ± 0.0000	-2 ± 0.0000	-1.8732 ± 0.3282	-1.9349 ± 0.1991
10	5	10	100	-2 ± 0.0000	-2 ± 0.0000	-1.9442 ± 0.1980	-1.8675 ± 0.3683
10	5	10	200	-2 ± 0.0000	-1.9994 ± 0.0074	-1.9406 ± 0.1150	-1.8437 ± 0.5275
10	10	1	50	-2 ± 0.0000	-1.999 ± 0.0117	-1.9836 ± 0.0798	-1.9108 ± 0.3004
10	10	1	100	-2 ± 0.0000	-2 ± 0.0000	-1.9894 ± 0.0942	-1.9357 ± 0.3001
10	10	1	200	-2 ± 0.0000	-1.9708 ± 0.3417	-1.9707 ± 0.1755	-1.938 ± 0.2311
10	10	5	50	-2 ± 0.0000	-2 ± 0.0000	-1.9058 ± 0.2896	-1.9524 ± 0.1740
10	10	5	100	-2 ± 0.0000	-2 ± 0.0000	-1.8685 ± 0.4217	-1.8662 ± 0.3748
10	10	5	200	-2 ± 0.0000	-2 ± 0.0000	-1.9888 ± 0.0513	-1.9396 ± 0.3280
10	10	10	50	-2 ± 0.0000	-2 ± 0.0000	-1.8584 ± 0.4222	-1.923 ± 0.2604
10	10	10	100	-2 ± 0.0000	-2 ± 0.0000	-1.9286 ± 0.4498	-1.7926 ± 0.4042
10	10	10	200	-2 ± 0.0000	-1.9985 ± 0.0126	-1.9801 ± 0.0500	-1.8642 ± 0.6628

G.4 TALKIE TRAINING COMPARISON

Here we provide the comparison of visualisation of Knowledge Anchor under different training regimes and their respective parameters.

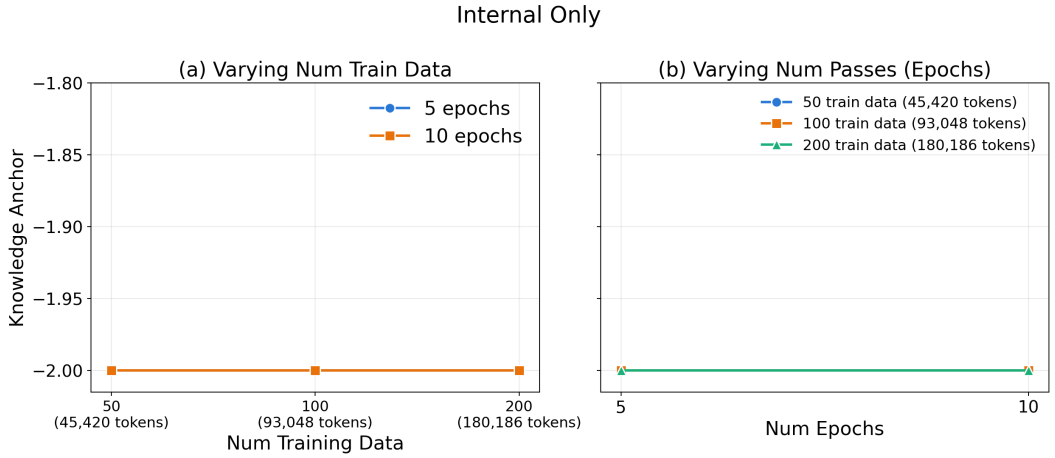

Figure 36: We analyse the effects on Knowledge Anchor for *Internal Only* when we vary (a) the number of training data (b) the number of data passes (epochs). Here, we only perform supervised fine-tuning using QA data on Talkie.

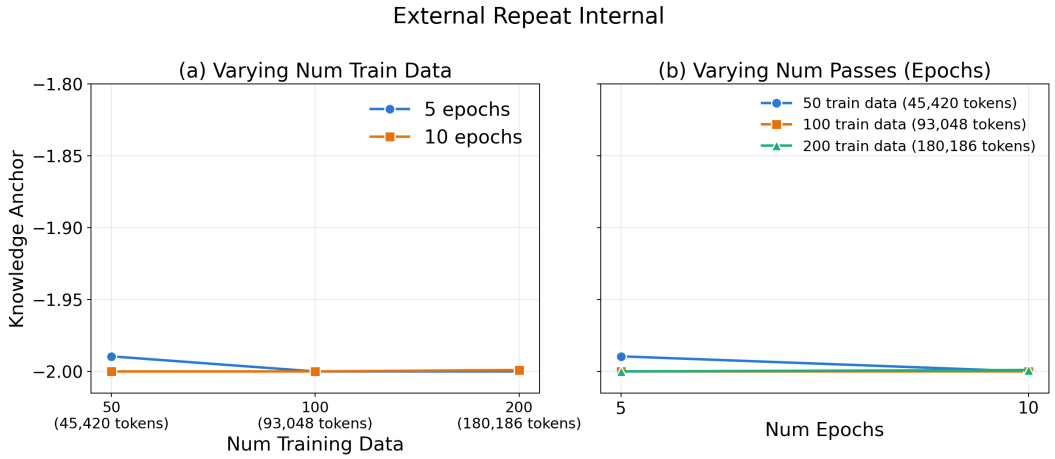

Figure 37: We analyse the effects on Knowledge Anchor for *External Repeat Internal* when we vary (a) the number of training data (b) the number of data passes (epochs). Here, we only perform supervised fine-tuning using QA data on Talkie.

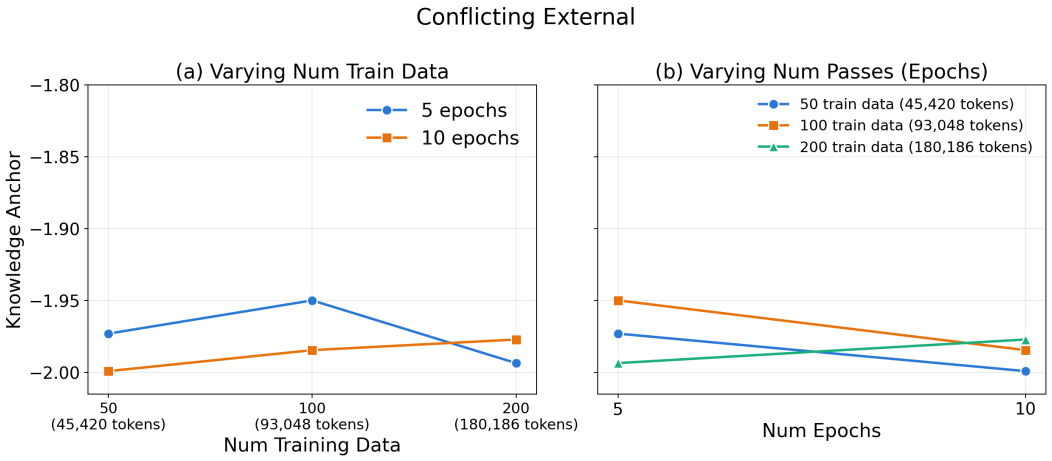

Figure 38: We analyse the effects on Knowledge Anchor for *Conflicting External* when we vary (a) the number of training data (b) the number of data passes (epochs). Here, we only perform supervised fine-tuning using QA data on Talkie.

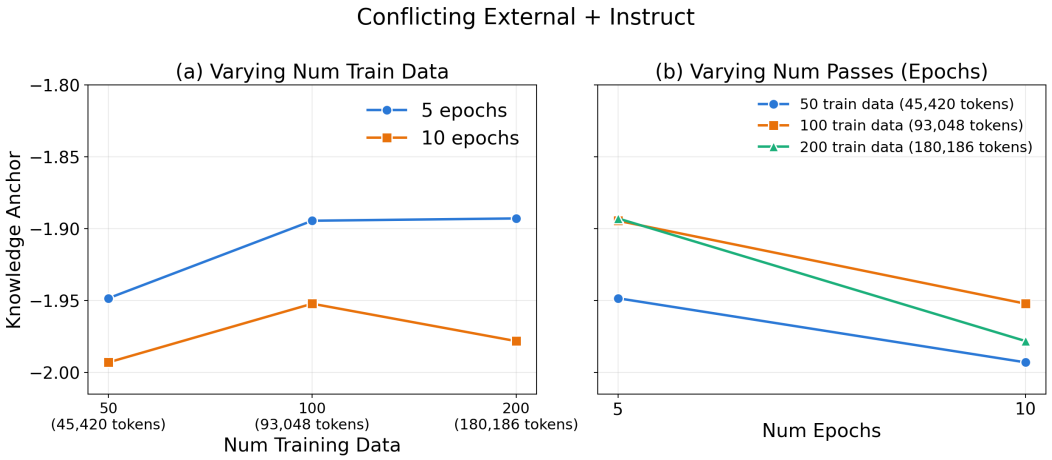

Figure 39: We analyse the effects on Knowledge Anchor for *Conflicting External + Instruct* when we vary (a) the number of training data (b) the number of data passes (epochs). Here, we only perform supervised fine-tuning using QA data on Talkie.

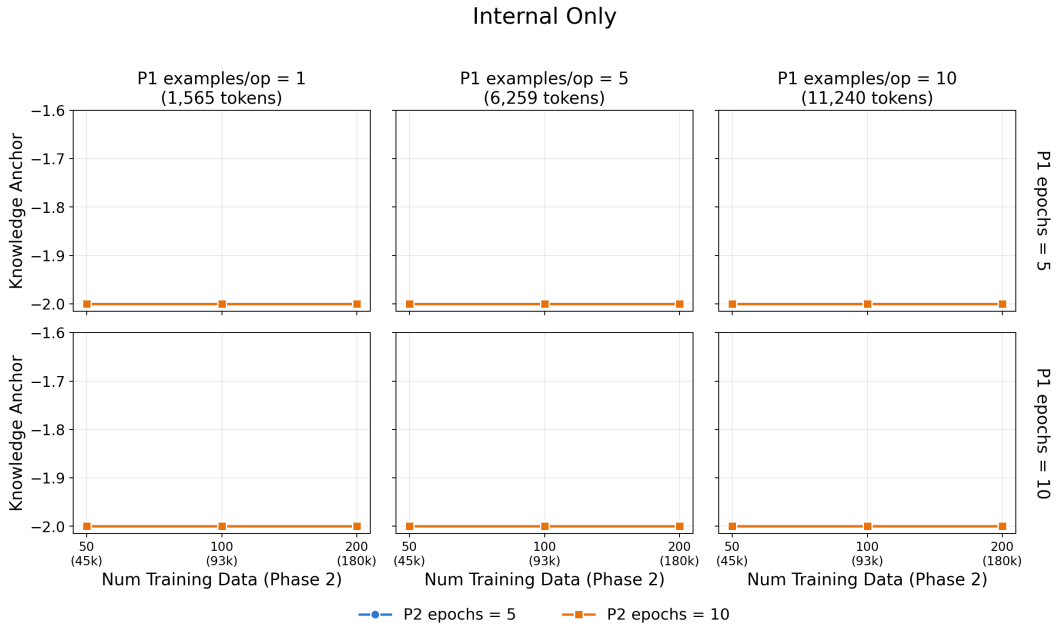

Figure 40: We analyse the effects on Knowledge Anchor for *Internal Only* when we vary (a) the number of operator sample use-case per documentation in Phase 1 (continued pre-training on SQL documentation) (b) the number of data passes (epochs) in Phase 1 (c) the number of training data in Phase 2 (LoRA supervised fine-tuning on QA data) (b) the number of data passes (epochs) in Phase 2.

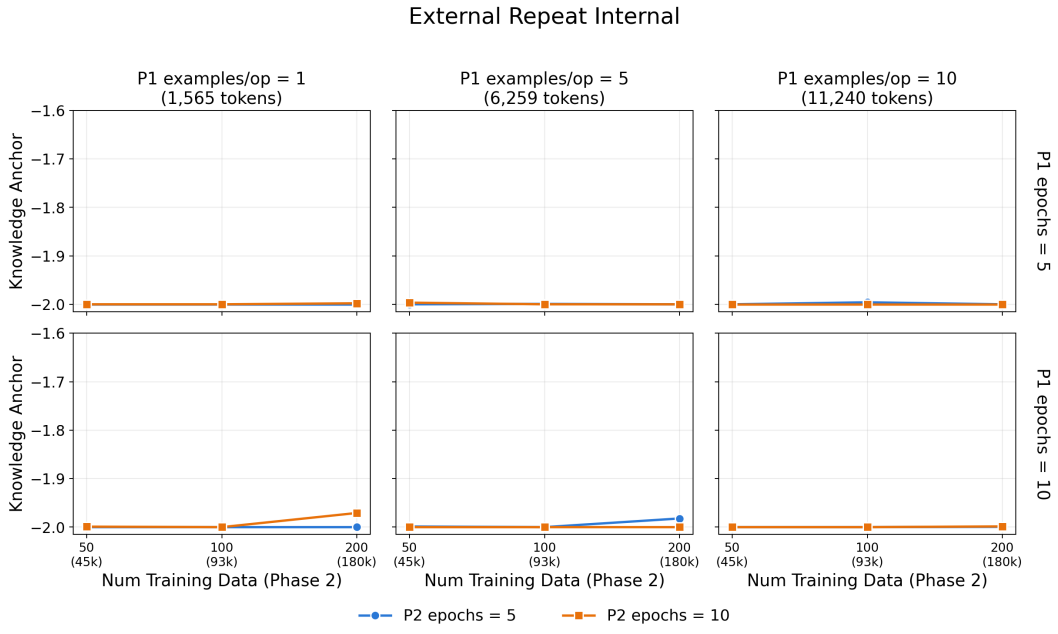

Figure 41: We analyse the effects on Knowledge Anchor for *External Repeat Internal* when we vary (a) the number of operator sample use-case per documentation in Phase 1 (continued pre-training on SQL documentation) (b) the number of data passes (epochs) in Phase 1 (c) the number of training data in Phase 2 (LoRA supervised fine-tuning on QA data) (b) the number of data passes (epochs) in Phase 2.

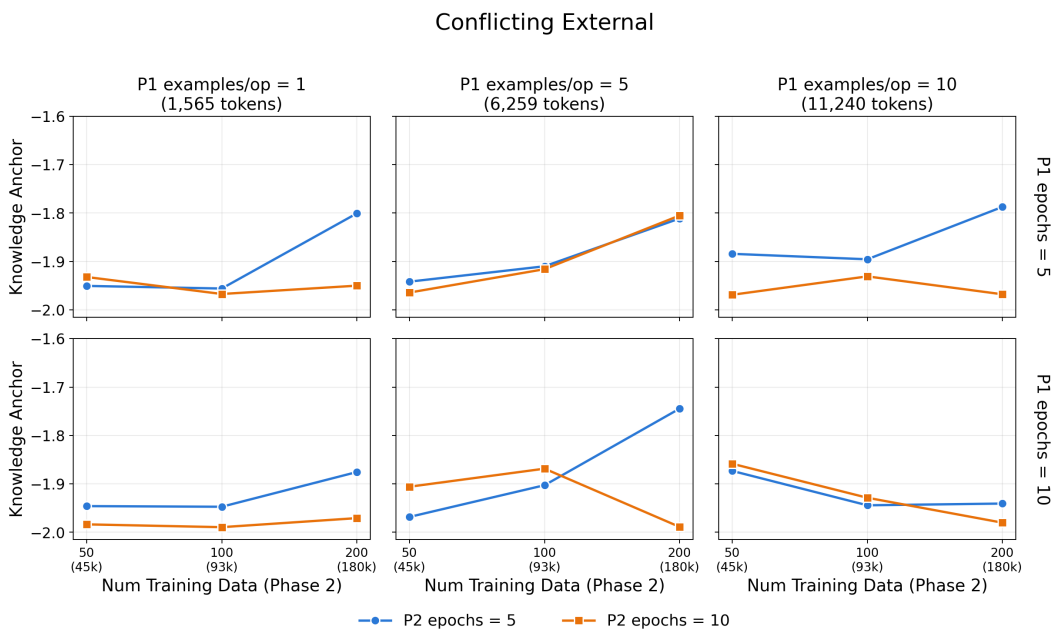

Figure 42: We analyse the effects on Knowledge Anchor for *Conflicting External* when we vary (a) the number of operator sample use-case per documentation in Phase 1 (continued pre-training on SQL documentation) (b) the number of data passes (epochs) in Phase 1 (c) the number of training data in Phase 2 (LoRA supervised fine-tuning on QA data) (b) the number of data passes (epochs) in Phase 2.

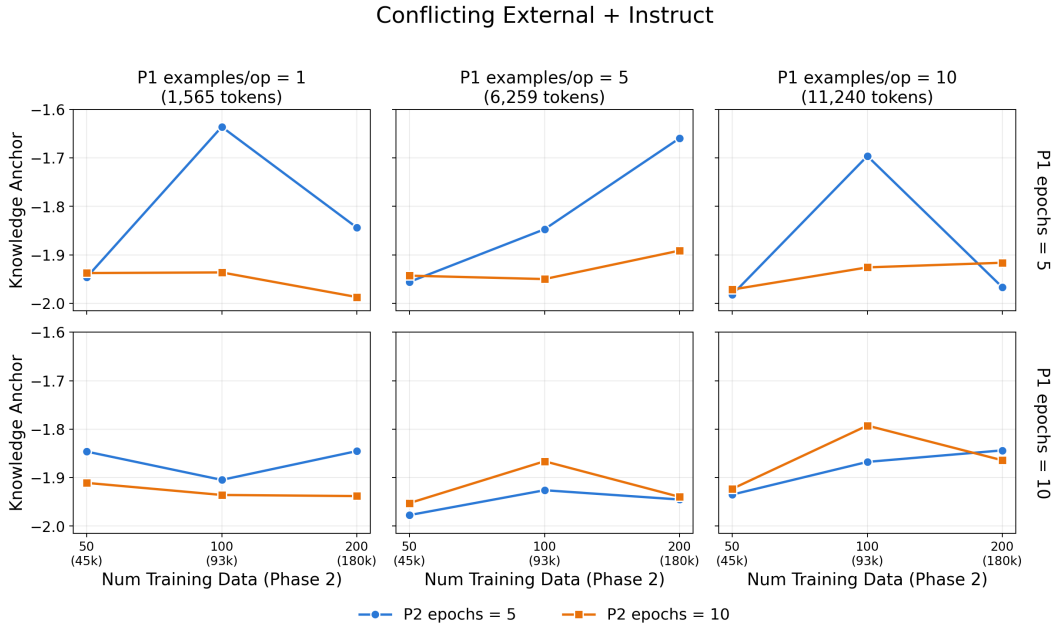

Figure 43: We analyse the effects on Knowledge Anchor for *Conflicting External + Instruct* when we vary (a) the number of operator sample use-case per documentation in Phase 1 (continued pre-training on SQL documentation) (b) the number of data passes (epochs) in Phase 1 (c) the number of training data in Phase 2 (LoRA supervised fine-tuning on QA data) (b) the number of data passes (epochs) in Phase 2.

H Extended Related Work

This appendix reproduces in full the literature review from which the condensed related work in the main paper was drawn.

H.1 Terminology and Scope

The terms memory, knowledge, experience, and even content remain loosely defined in the current literature. In particular, the exact definition of memory is unclear: it can be defined as the container/repository that stores knowledge, information or skills, or it can be defined as a more general process encompassing encoding and storage of knowledge, or it can also be used to refer to the knowledge contents themselves. Xu et al. (2024); Gu et al. (2026), when investigating knowledge attribution under conflict, have used the term memory and knowledge interchangeably, not distinguishing any difference between the two in the context of knowledge attribution. On the other hand, Wu et al. (2025) defined memory as not only the process of encoding, storing, and retrieving information but also the ability to do so.

In this work, we follow Wu et al. (2025) and use memory to define the container in which pieces of knowledge are stored. It is thus the repository which encloses the knowledge which a model's reasoning is attributed to.

H.2 LLM Memory and Knowledge

Memory is a crucial capability of LLM-based agents, responsible for storing information and relevant content to support decision-making. It can be viewed as the repository of all knowledge concepts that the model has come across during training, or it can also be provided externally. As such we can typically classify LLM memory and therefore knowledge into two main forms: internal knowledge, which refers to the knowledge stored directly in the models' parameters, and external knowledge, which are additional inputs/databases that provide further information not embedded within the models' parameters. Hence, we can also refer to internal knowledge and external knowledge as parametric and non-parametric knowledge respectively Zhang et al. (2026).

H.2.1 Internal knowledge: acquisition, shaping and editing

Internal knowledge in LLMs is parametric in nature: it is encoded in the model's weights and updated through training (Zhang et al., 2026; Yin et al., 2024). As such, acquiring internal knowledge often involves extensive training on a large corpus. In initial pre-training, basic knowledge on language, semantics and the ability for next-token text completion is acquired by the model. Continued pre-training and supervised fine-tuning can be applied for the model to gain further domain-specific knowledge. While these techniques typically require updating the entire model, memory-efficient training methods such as LoRA Hu et al. (2022) can instead be used to encode internal knowledge while updating only a small fraction of the model's parameters.

Acquisition through pre-training The base form of internal knowledge is acquired during pre-training, where model parameters are trained to maximize the likelihood of the textual data they are exposed to (Liu et al., 2025). When given tokens $w_1, \ldots, w_{t-1}$ and T training data, the model's objective is to predict the next token w_t, which gives us the following loss function:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} - \log P(w_t \mid w_1, w_2, \ldots, w_{t-1}) \tag{7}$$

The emergence of parametric knowledge has been observed to correlate strongly with the MLP sub-layer outputs, where they serve as key-value memories: the first linear layer's parameters are keys that capture input patterns and the second layer's parameters capture the distribution. In fact, lower feed-forward layers are found to capture shallow single-word patterns, while upper layers learn more semantic, topic-level patterns Geva et al. (2021b). In the MHA block, different attention heads are also found to be specialized in different contexts Zhao et al. (2025).

Continued pre-training (CPT) Beyond the initial pre-training stage, internal knowledge can be further shaped through continued pre-training, where the additional datasets are focused on a particular domain Siriwardhana et al. (2024). CPT uses the same set-up as the initial pre-training of the model, with the text-completion objective now optimised on the new domain-specific corpora Gupta et al. (2023). While this is often chosen as an initial method to enhance a model's knowledge in a specialized area Lu et al. (2025), a lack of an appropriate training design could also lead to

catastrophic forgetting where fine-tuned models lose their previous fine-tuned abilities Li & Lee (2024).

Supervised fine-tuning (SFT) While CPT serves to inject domain-specific knowledge to the model's internal knowledge, the model was only trained with a text-completion objective and hence is typically still unable to complete domain-specific tasks. As such, supervised fine-tuning (SFT) is adopted as a second training phase after CPT. SFT departs from CPT in that it trains on labelled data such as question-answer pairs, teaching the model in a supervised manner to utilise the domain-specific knowledge to answer questions Lu et al. (2025); Liu et al. (2025).

Low-Rank Adaptation (LoRA) While full CPT and SFT train on all of the model's parameters, LoRA seeks to reduce the computational and memory overhead of doing so by training on a small subset of parameters. In LoRA, the model's original weights are frozen and instead a smaller, lightweight low-rank matrix is added, and SFT is performed on this low-rank matrix instead Joshua Noble (2026); Hu et al. (2022). This has been found to reduce trainable parameters by thousands of times, reducing the overall time and cost. With LoRA, a single model can be trained to have multiple specialities by simply training multiple LoRA adaptors. The adaptors can be toggled on and off, allowing the model to revert back to the initial version. However, with learning fewer parameters, a trade-off is that the model tends to learn less. Biderman et al. (2024) found that models trained with LoRA underperform full fine-tuned models in code and mathematics but at the same time suffer from lesser forgetting issues than in full fine-tuning.

Knowledge editing In knowledge editing, instead of general model training, it locates and modifies specific parameters in the model responsible for a given factual association, leaving the rest of the model untouched Wang et al. (2024). This method serves as a small-scale knowledge update where an original piece of knowledge is updated to a new version. Meng et al. (2022) introduced ROME which treats an MLP weight matrix as a key-value associative memory similar to that introduced by Geva et al. (2021b). It inserts a new fact with a closed-form rank-one update by modifying the MLP parameters just enough to map a new key to a new value while keeping other keys unchanged.

H.2.2 External knowledge: in-context, retrieval, tool calling and external memory models

As alluded by the term non-parametric knowledge, external knowledge refers to information that is provided to the model through external means, i.e. it is not embedded within the model's internal parameters. As such, knowledge through external memory is a great way to store personal data, or to obtain information that the model was not trained on Zhou et al. (2026); Zhang et al. (2025b).

In-context A direct method of supplying external information is simply placing the said information directly in an LLM's context window. Experiments have shown increased capabilities of LLMs through in-context learning where example tasks written in natural language are included in the prompt, along with the query task Dong et al. (2024); Brown et al. (2020). However, multiple evaluations also report that evidence placed in the middle of an ultra-long sequence is harder for the model to utilize effectively than evidence near the beginning or end, an effect that persists even as the context window is enlarged (Zhang et al., 2025a; Liu et al., 2024).

Retrieval-Augmented Generation (RAG) Another common method is using Retrieval-Augmented Generation (RAG) systems, which are typically built on top of two components: a storage layer and a retrieval layer Lewis et al. (2020); on Hugging Face documentation using LangChain (2026). There are multiple ways to represent text and hence many different storage methods. One approach is vector-based, which is similar to representing text as embeddings. A piece of text can also be decomposed to individual pieces of information allowing them to have structural representations using knowledge graphs. Most retrieval methods are either lexical-based, semantic-based or a combination of both. Lexical-based retrieval such as BM25 Robertson & Zaragoza (2009) or Jaccard similarity measures the level of overlap between the target query and the documents stored. These methods consider the query and documents as individual sets, then measure the frequency of common tokens in each set, retrieving the documents with highest overlap. In semantic-based retrieval, the similarity between the vectorized query and documents is measured using a distance metric such as cosine similarity (Wu et al., 2026b). The retrieval methods typically retrieve the top k similar documents, supplementing them to the LLM in-context to solve the target task. While the individual retrieval methods serve as a good baseline, hybrid methods which combine both semantic and lexical retrieval using Reciprocal Rank Fusion (RRF) have been found to outperform the individual methods Cormack et al. (2009). By

combining both metrics, RRF is able to both capture semantic similarity and frequency of keywords between the query and documents. Other hybrid methods such as decomposing a query into keywords then retrieving using both the keywords and full query have been found to be useful in identifying documents that capture both the general topic and individual details in the query Guo et al. (2025).

Tool calling A separate line of work equips agents with external tools — databases, or code interpreters exposed through APIs Stryker (2026). This requires the LLM to recognize that it lacks the necessary capabilities to complete the task but understands that it is equipped with the tools that it can utilise to do so Qin et al. (2024). Tool-calling in LLMs involves the model providing the external tool with the necessary information, executing the tool and using the output generated to complete the task.

External memory model Beyond the methods listed above, several architectures organize memory and thus knowledge using complex structures. Quek et al. (2026) introduced MeMo which combines both parametric and non-parametric methods to encode LLM knowledge. They do so by encoding the target corpus into a dedicated Memory model, then let any frozen LLM query the Memory model at inference time through a multi-turn protocol where the frozen LLM reasons over the documents retrieved. The MeMo architecture splits knowledge and reasoning into two components: a MEMORY model, which contains the new knowledge, and a frozen EXECUTIVE model, which functions as the reasoning engine. This method thus leverages the strength of both parametric and non-parametric methods while reducing their individual limitations.

H.2.3 KNOWLEDGE REMOVAL, FORGETTING AND MACHINE UNLEARNING

While acquiring more up-to-date and diverse data remains important under the "bigger is better" paradigm, growing attention has recently been paid to ensuring that models do not retain harmful information. To address this, a new class of methods known as model unlearning has emerged, aimed at removing the influence of undesirable data from LLMs. Unlearning methods have been widely used in non-textual tasks such as image classification Ali et al. (2025). A straightforward way to achieve approximate unlearning is to simply use gradient ascent. As opposed to gradient descent, which is used in most machine learning training to minimize the objective loss, gradient ascent seeks to maximize the loss instead Jang et al. (2023). While the loss is the same as before,

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} -\log P(w_t \mid w_1, w_2, \ldots, w_{t-1}) \tag{8}$$

gradient ascent applies it to the model parameters θ in the opposite direction:

$$\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L} \tag{9}$$

This reduces the probability that a model reproduces that exact sequence of tokens, essentially unlearning it.

While simple gradient ascent provides an intuitive baseline for unlearning methods, it however suffers from a few issues:

1. Unlike gradient descent, the loss is unbounded in gradient ascent. As such, too many rounds of unlearning may lead to catastrophic forgetting of other capabilities of the model Huang et al. (2026); Zhang et al. (2024).
2. Knowledge in LLMs is entangled Chen et al. (2026), making it almost impossible to isolate a particular set of knowledge from others Gong et al. (2025). For example, changing the capital city of the United Kingdom from London to Washington D.C. will also require changing the fact that there is no city named Washington D.C. in the United Kingdom. Another problem associated with unlearning is that while the model achieves the objective to unlearn the target forget dataset, it may lose knowledge in other desirable areas as well Zhang et al. (2024); Maini et al. (2024).

Many modifications to the baseline are thus proposed to address this issue. Where possible, a separate retain dataset containing knowledge to be retained is also used in unlearning. Yao et al. (2024) proposed a composite loss which minimizes the loss on the retain dataset while concurrently maximizing the loss on the forget dataset. Zhang et al. (2024) proposed Negative Preference Optimization, a modification of the Direct Preference Optimization loss Rafailov et al. (2023), keeping only the "lose" preference, which thus reduces prediction probability on the forget set while keeping the loss bounded.

H.3 Conflicts in Knowledge

Given the important role that different forms of knowledge play in shaping model behaviour, a crucial question arises: what happens when these memories diverge on a given topic? Internal and external knowledge need not always agree: they may reinforce one another when aligned, complement each other when they provide non-overlapping information, or conflict when the evidence they provide is contradictory Cheng et al. (2024). In the last case, where internal and external knowledge sources disagree, this poses the greatest challenge, since the model must implicitly decide which source to trust and use in output generation.

H.3.1 Internal vs. Internal Knowledge Conflict

This form of conflict arises from inconsistencies within a model's own parametric knowledge, leading to inconsistent responses to semantically equivalent inputs and, consequently, unpredictable model behaviour. Xu et al. (2024) identified three main sources of such a form of conflict:

1. **Conflicting or contradicting pre-training data.** Because models are trained on massive corpora spanning diverse topics and sources, individually verifying or cleaning each piece of data is often not feasible, making conflicting information difficult to avoid within the training corpus (Jia et al., 2026).
2. **Knowledge editing.** Knowledge editing inherently alters the model's next-token distribution and therefore its output. In particular, editing facts that are logically inconsistent with each other can introduce inconsistencies elsewhere in the model (Li et al., 2024).
3. **Stochastic output decoding.** Since the output of the model is based on a probability distribution, non-greedy decoding methods mean that the output in each round of probing is non-deterministic and could be very different. This is particularly true in contexts where the model is highly uncertain.

Pham et al. (2026) took a closer look at how conflicting knowledge is encoded in LLMs' parameters by continually pre-training two models, l'_{mix} and l'_{clean}, on conflicting and non-conflicting data respectively. Using activation patching, they find that the model's final layers are the predominant levers governing changes in token probabilities, indicating that higher layers play a more dominant role in managing internal knowledge conflict.

H.3.2 External vs. external knowledge conflict

This form of conflict occurs when the external knowledge retrieved is contradictory, and is due to two main sources:

1. **Misinformation** occurs since most RAG systems do not have the ability to distinguish trustworthy sources from non-trustworthy ones, or factual information from counterfactual ones. When data is sourced from web sources such as a forum, it could contain many opinion-based texts which could contain fake news. When RAG queries for relevant information, it simply seeks semantically similar documents and does not identify misinformation that may occur in the documents Xu et al. (2024).
2. **Temporal shift** occurs when facts change over time but, as documents are static, they are not updated; instead, this change is reflected in a new document. As such, there could be more than one document representing a particular topic, where one could be outdated.

Hou et al. (2024) constructed WikiContradict, a benchmark of naturally occurring contradictions extracted from Wikipedia's own maintenance tags, and found that LLMs struggled to identify the correct answer when presented with contradictory retrieved evidence. Rather than reconciling the discrepancy, some models simply latched onto a single passage while disregarding the rest, whereas others avoided the issue by listing every candidate answer without indicating that they were mutually exclusive. This effect was most pronounced for implicit contradictions requiring multi-step reasoning to detect. Wang et al. (2025) likewise showed that as the proportion of misinformation among retrieved documents increases, standard RAG pipelines degrade by as much as 46% in exact-match accuracy for some baselines, as a higher density of untrustworthy evidence erodes the model's ability to separate fact from fabrication. St-Charles et al. (2026) demonstrate an example where the documentation of a function conflict with what the function actually execute. In such cases, models sometimes follow the documentation instead of executing the code.

A range of approaches have since been proposed to address such knowledge conflicts. Lau et al. (2025) decomposed a user query into its semantic and temporal components, splitting retrieval into two stages: a temporal filter that first narrows the corpus to documents consistent with the

query's intended time frame, after which the system ranks the remaining candidates by semantic relevance. Logan (2026) pursue a related idea at the architecture level, building a continuum memory that attaches timestamps to stored documents so that outdated information can be systematically de-prioritized over time. Wang et al. (2025) introduce MADAM-RAG, a multi-agent framework in which each retrieved document is assigned to its own agent that reasons over it independently; the agents then debate across several rounds, iteratively revising their positions until they converge on a single, reconciled response. Wu et al. (2026a) took a training-based approach instead, teaching the retriever itself to flag distracting or contradictory documents and to attach a conflict-confidence score at the re-ranking stage, providing the LLM with an explicit signal for whether a query is safely answerable or likely to involve conflicting sources.

H.3.3 INTERNAL VS. EXTERNAL KNOWLEDGE CONFLICT

The third form of conflict arises when the model's internal knowledge conflicts with the external knowledge provided Xu et al. (2024). The cause of this is similar to the other two cases of conflict, where there could be misinformation or outdated information in either the internal or external knowledge. Internal vs. external knowledge conflict has been extensively studied and a few observations have been made:

There is no definitive conclusion on which type of knowledge LLMs favour more under conflict. Xie et al. (2024) tested knowledge conflict under a question-answering setting using datasets such as POPQA where entities in the original answer were swapped to counter versions to generate conflict. Xie et al. (2024) found that in such cases of simple entity-based substitution, the models tended to pick the original version from their internal knowledge, even when the models were instructed to use the knowledge provided in-context to answer the questions. But when the counterfactual external knowledge was generated in a coherent way by providing supporting evidence instead of pure substitutions, LLMs instead returned answers mostly consistent with external knowledge. Qian et al. (2024) took a knowledge graph approach to investigate knowledge conflicts, representing knowledge as graphical relations. External knowledge was similarly constructed by swapping entities in the knowledge graph with counterfactual versions, known as distractors. They observed that when the model was originally very confident in its internal knowledge, it remained conforming to its internal knowledge, unaffected by the external distractors, but also noted that when it was not initially confident in its internal knowledge, it was more likely to use the external knowledge. Cheng et al. (2024) similarly took on a question-answering approach using their self-created benchmark ECHOQA and created four different types of external evidence: supportive, complementary, conflicting and irrelevant, and instead found that most models do not typically lean on their internal knowledge under conflict.

LLMs are sensitive to how external evidence is presented, such as its ordering and frequency. Xie et al. (2024) experimented with varying the level of conflict in-context by changing the ratio of counterfactual evidence and original evidence supported by internal knowledge provided in-context. They noticed that LLMs displayed a herding behaviour, choosing to support the side with the greater number of supporting evidence. They also made the observation that LLMs displayed sensitivity to evidence ordering and tended to choose the side whose evidence was placed first. Jin et al. (2024) also noted that LLMs tend to choose the side with evidence appearing more frequently, but also made the secondary observation that excessively long context might cause the model to lose the focus of the query.

As the knowledge popularity increases, the model's utilization of its internal knowledge under conflict increases. Xie et al. (2024) tested knowledge conflict on the POPQA dataset which was derived from WIKIDATA, which also provides a popularity measurement of the question, based on monthly Wikipedia page views of the entity associated with the question. They found that as the popularity of the question increases, the LLMs displayed increased utilization of their internal knowledge to as much as 80%. Cheng et al. (2024) echoed this observation and noted that utilization of internal knowledge for popular questions is almost twice as high as for less popular ones. Jin et al. (2024) similarly also observed that as the popularity of the question decreases, LLMs tended to trust external knowledge more.

H.4 Evaluating internal vs. external conflicts

A central problem in evaluating the performance of models under knowledge conflict is designing proper metrics as well as datasets that reliably isolate the model's reliance on internal versus external knowledge. Here we introduce some of the methods adopted by prior works.

H.4.1 Metrics

Memorization One common way to measure the tendency of models to stick to internal knowledge over external knowledge under conflict is using the memorization ratio, which simply measures how often the model sticks to its internal knowledge as compared to the provided external context Xie et al. (2024); Jin et al. (2024); Chen et al. (2022); Longpre et al. (2021). This requires a dataset where external knowledge and internal knowledge are mutually exclusive such that when the model outputs a particular response, one is able to attribute it to the correct knowledge source. The memorization ratio is given as follows, where f_i indicates the frequency of the model outputting a response from internal knowledge and f_e the frequency of the model outputting a response from external knowledge:

$$\text{Memorization Ratio} = \frac{f_i}{f_e + f_i} \tag{10}$$

Knowledge preference A more direct way than memorization is to simply measure the knowledge preference of a model, where preference for internal knowledge is given by the frequency of the model providing an answer from internal knowledge and vice versa for external knowledge. The external knowledge is typically constructed counterfactual facts, so they are mutually exclusive from the internal knowledge. Kim et al. (2026) tested this using a synthetic biography dataset and constructed the following metrics:

$$\text{Pref}_i = \mathbb{E}_{e\sim\mathcal{E}_{\text{train}}}\left[\frac{1}{|A_e|}\sum_{a\in A_e}\mathbf{1}\{M(C'_e, p_a) = v_a\}\right] \tag{11}$$

$$\text{Pref}_e = \mathbb{E}_{e\sim\mathcal{E}_{\text{train}}}\left[\frac{1}{|A_e|}\sum_{a\in A_e}\mathbf{1}\{M(C'_e, p_a) = v'_a\}\right] \tag{12}$$

Pref_i gives the preference for internal knowledge and Pref_e gives the preference for external knowledge. Kim et al. (2026) tested, when given an entity e, test probe p_a and a context C_e where attributes of e are replaced with a randomly selected value, does the model M return the correct attribute value v_a or the randomly sampled version v'_a.

Confidence The above metrics give direct indications of a model's preference between internal and external knowledge under knowledge conflicts. Since LLMs are auto-regressive models, experiments by Jin et al. (2024); Xie et al. (2024); Qian et al. (2024) have considered the model's response from the probability distribution lens by considering the deviation in probability of the answer as compared to under pure internal knowledge. The probability of the model giving an answer is thus viewed as the confidence of the model, where a higher probability indicates greater confidence and vice versa.

H.4.2 Datasets

Question-answering datasets are typically used in conflict detection/resolution. These are typically secondary datasets derived from pre-existing QA datasets by generating a corresponding counterfactual version. Cheng et al. (2024) used ConflictQA developed by Xie et al. (2024), which combines a WikiData question-answering dataset PopQA Mallen et al. (2023) and a multi-hop reasoning question-answering dataset StrategyQA Geva et al. (2021a). Cheng et al. (2024) developed EchoQA which combines various datasets encompassing scientific, factual and common-sense knowledge and generated counterfactual versions of these datasets which serve as the external knowledge. Wu et al. (2024) took a more comprehensive approach in investigating conflict by introducing 11 counterfactual versions of real-world tasks including code generation, drawing and music. These tasks involve swapping out real-world versions of the tasks to something different such as drawing an image upside down, or sorting a list by the second element instead of the first element.

Other work has focused on probing whether LLMs can generalise knowledge they already possess to settings they are unfamiliar with. Sharma & Chopra (2026) introduced EsoLang-Bench, a benchmark of programming problems which covers coding tasks of varying difficulties, and models are required to solve them using esoteric programming languages which are rare and not commonly used and are thus under-represented in most training corpora. They tested on 5 frontier models

and found that the models were not able to generalise their coding abilities, achieving only 0–11% accuracy on the esoteric programming languages but able to achieve 100% on the same questions using popular languages like Python and JavaScript. Zheng et al. (2026) investigated the ability of LLMs with NEWTONBENCH to make scientific discoveries by generating counterfactual scientific laws spanning across 12 domains. Each model was provided with assisting equations and had to interact with a simulation to derive the target shifted scientific law. Ma et al. (2025) developed KOR-BENCH which similarly tests the model's ability to elicit applicable relevant knowledge and follow rules of a never-before-seen task and solve problems spanning logical, operator, cipher, puzzle, and counterfactual reasoning. These tasks require the model to be able to identify underlying relevant knowledge that it possesses, be able to generalise to new contexts and follow the requirements provided in-context, and not simply memorize what it has been trained on previously.