REASONING AND LEARNING IN TEXT-TO-SQL: FROM TEMPLATE IMITATION TO STRATEGY RECOMBINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models increasingly use chain-of-thought (CoT) reasoning to translate natural-language questions into SQL queries. Existing methods provide diverse reasoning templates, but how models use these templates after supervised fine-tuning and reinforcement learning remains unclear. In particular, can learning from multiple templates lead to useful reasoning beyond the imitation of a single template? In this work, we empirically study this transition on BIRD using two 7B models. We construct matched training data with balanced coverage of four templates and compare supervised fine-tuning, rejection fine-tuning, and subsequent GRPO. By analyzing freely generated responses, intervening on requested templates, and comparing reasoning functions before and after RL, we find that balanced training does not produce balanced template use in greedy outputs. Instead, RL changes both the dominant forms of expression and the organization of reasoning functions within complete responses. Among same-question greedy responses with identifiable reasoning functions, RL changes their combination or order in 78.74% and 73.11% of cases for the two SFT models. Reasoning organizations associated with successful solutions become more frequent after RL across all four supervised training paths, including when these organizations are identified using pre-RL responses alone. These findings show how multi-template learning and RL can support a shift from template imitation toward task-useful CoT organization, providing an empirical basis for studying strategy recombination in text-to-SQL.

1 INTRODUCTION

Chain-of-thought templates provide explicit examples of how to approach a task: decompose a question, resolve schema references, plan a query, or revise an unsuccessful attempt. Exposing a language model to several such templates raises a learning question beyond final-answer accuracy: does post-training reinforce particular demonstrated trajectories, or enable it to organize useful CoT without following one complete template? A template specifies both reasoning content and an order of presentation. Learning from several templates may therefore provide alternative ways to express similar task-relevant functions. The central question is how subsequent reward-based training changes the use of these alternatives when no particular template is requested.

We study this question in text-to-SQL, using controlled experiments and supplementary scaling results on BIRD (Li et al., 2023). Execution provides a common outcome measure, while alternative reasoning traces let us compare the expression accompanying correct and incorrect answers. A model may become more accurate by concentrating probability on familiar successful answers, without expanding the questions it can solve. Candidate coverage and greedy accuracy can consequently favor different training paths under identical evaluation conditions. We distinguish three training methods: SFT on execution-verified teacher demonstrations, RFT on the model's own execution-verified generations, and RL using an execution-related reward. RFT applies the supervised objective to correctness-filtered trajectories, separating the effect of target construction from the optimization loss. The primary comparison uses two 7B backbones with matched training-question identities and template-source counts; supplementary 32B results provide a separate scaling perspective.

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.

Our central hypothesis is that multi-template experience can become a resource for more general CoT expression under RL. We define *strategy recombination* behaviorally: moving beyond rigid reproduction of one complete template toward freely organized, task-useful reasoning. We characterize general CoT through the functions expressed in complete responses, including forms without explicit template labels. The comparison measures how RL changes the use of useful organizations available after supervised learning. This requires connecting three observations: a change in expression, a change in the organization of reasoning functions, and an association with successful solutions. Comparing the same questions before and after RL links these observations while keeping the task fixed. In particular, a model may retain a familiar opening while changing the functions expressed later in its answer. Conversely, a new opening may accompany a similar reasoning sequence. We therefore analyze both recognizable formats and complete prose, and evaluate useful organizations on candidates separate from those used to identify them.

Our analysis links template use, reasoning organization, and solution quality. We compare freely generated formats with responses to supplied scaffolds, measure changes in the functions expressed throughout each answer, and evaluate the accuracy of increasingly frequent organizations on held-out candidates. This progression connects observable expression changes with their practical value for SQL generation. We distinguish identifiable template markers from SQL-block openings and unmatched formats, and count incomplete answers separately. Our detectors measure surface markers and the functions expressed throughout complete reasoning traces. These comparisons link expression and organization to execution quality under a shared evaluation protocol.

Our contributions are:

- A matched study of two 7B backbones across original, SFT, RFT, and GRPO states, with development questions excluded from training and checkpoint selection.
- Linked tests of template dependence, free CoT organization, and held-out utility, including pre-RL selection controls that associate SFT-derived organizations with held-out accuracy.
- Auxiliary likelihood and cross-backbone analyses, complemented by supplementary 32B results on the relationship between answer reliability and sampled coverage.

2 Related Work

Reasoning templates and selection in text-to-SQL. DIN-SQL decomposes SQL generation into subproblems with self-correction (Pourreza & Rafiei, 2023); CHESS combines retrieval, schema selection, candidate generation, and validation (Talaei et al., 2024); E-SQL enriches questions with relevant schema information (Caferoğlu & Ulusoy, 2024). These methods structure intermediate reasoning around decomposition, schema information, and validation before producing the final query. CHASE-SQL combines multiple reasoning paths with a preference-trained selector (Pourreza et al., 2025). More recent work, R^3-SQL, explicitly groups candidates by execution results and combines ranking with resampling (Han et al., 2026). We examine post-training changes within a single model's freely generated responses, complemented by requested-scaffold comparisons.

Learning from verified generations. STaR iteratively trains on successful rationales and uses answer-conditioned rationalization when initial generations fail (Zelikman et al., 2022). Arctic-Text2SQL-R1 demonstrates the effectiveness of execution-based RL for SQL generation (Yao et al., 2025). In an empirical study of RL with verifiable rewards, Yue et al. (2025) find that gains at small sampling budgets can coexist with lower coverage at larger budgets. Their comparison treats the original model's candidate distribution as a reference for assessing post-RL reasoning capacity. We extend this reliability–coverage perspective to reasoning organization after multi-template SFT and RFT.

Multiple samples and answer aggregation. Pass@k measures whether a set of samples contains a successful solution (Chen et al., 2021), while self-consistency aggregates generated answers to choose one output (Wang et al., 2023). For SQL, different strings may return the same result, and normalization policies can change voting groups. These approaches separate the availability of a correct candidate from the ability to select it. Pass@k counts a question as covered whenever at least one candidate succeeds, whereas aggregation evaluates the answer selected from that pool.

3 EMPIRICAL STUDY

3.1 EXPERIMENTAL SETUP

We compare six states of Qwen2.5-Coder-7B-Instruct and Qwen2.5-7B-Instruct (Hui et al., 2024; Yang et al., 2024): original, SFT, RFT, and standard GRPO (Shao et al., 2024) continued from each of these three initializations. SFT and RFT independently start from original weights. The fixed cohort has 5,424 BIRD training questions, each with one target under CHESS, DIN-SQL, E-SQL, and Plan (21,696 targets per source). The first three templates adapt the named methods; Plan uses execution-plan-style instructions. Both SFT models use execution-verified Qwen2.5-Coder-32B-Instruct outputs; RFT uses each backbone's own execution-verified generations. GRPO uses the same ordinary question prompts. All ten training stages use seed 0 and predetermined final checkpoints.

SFT and RFT run for three epochs at a learning rate of 2×10^{-5}, with a global batch size of 64 and 1,017 optimizer updates. GRPO runs for one episode at a learning rate of 10^{-6}, using 16 prompts with eight candidates each per rollout, for 339 rollouts.

We evaluate all 1,534 development questions, excluded from training, rejection filtering, and checkpoint selection, using ordinary question prompts and a maximum of 2,048 new tokens per answer. Sampled decoding uses 32 candidates per question at temperature 0.8 and top-p 0.95. Greedy execution accuracy (EX) measures one deterministic answer; sampled Pass@k estimates success within k of 32 candidates as $1 - \binom{32-c_i}{k} / \binom{32}{k}$, averaged over questions with c_i correct candidates. Pass@32 measures full-pool coverage; rate differences use $\Delta = r_A - r_B$ in %. Format failures, execution errors, timeouts, and truncations remain in the denominator. Appendix A provides construction, optimization, token-budget, and implementation details.

3.2 SUPERVISION AND THE SINGLE-ANSWER–COVERAGE TRADE-OFF

We ask whether the supervised path preferred for a single answer also provides the broadest sampled coverage. We compare SFT and RFT within each backbone on the same 1,534 development questions, using predetermined final checkpoints and identical decoding and scoring protocols. The comparison covers greedy accuracy and sampled success at increasing candidate budgets. The curves distinguish individual-answer reliability from question coverage at larger sample budgets.

Table 1: Primary 7B results on all 1,534 development questions (percent). Greedy EX is separate from sampled Pass@k, estimated from 32 candidates per question. Each of the twelve model states is evaluated with greedy and sampled decoding.

		Greedy	**Sampled Pass@k**			
Model	**Method**	**EX**	$k = 1$	$k = 8$	$k = 16$	$k = 32$
Coder	Original	52.15	48.92	71.28	75.76	79.14
Coder	SFT	59.06	58.49	73.03	75.48	77.51
Coder	RFT	53.39	52.27	74.46	78.47	81.55
Coder	Original + GRPO	58.47	58.47	65.53	66.79	67.80
Coder	SFT + GRPO	59.71	59.54	73.09	75.26	76.99
Coder	RFT + GRPO	55.48	54.73	74.54	78.19	81.23
General	Original	41.79	41.44	54.74	57.73	60.04
General	SFT	55.48	54.61	71.58	74.60	76.86
General	RFT	49.35	47.67	69.48	73.59	76.86
General	Original + GRPO	53.85	53.09	60.73	62.62	64.15
General	SFT + GRPO	56.65	56.04	71.39	73.99	75.95
General	RFT + GRPO	52.09	51.56	68.25	71.49	74.12

The sample-budget curves compare these supervised paths as more candidates become available, showing whether their relative ordering changes beyond single-answer performance.

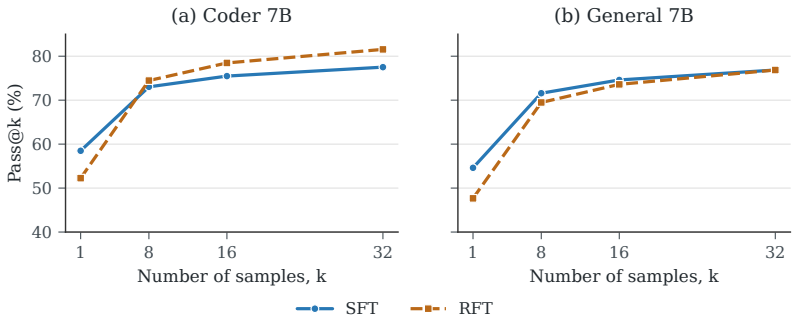

Figure 1: 7B SFT and RFT sample-budget curves on the same 1,534 questions. Markers show Pass@k estimates at the evaluated candidate budgets, calculated from 32 samples per question.

SFT favors single-answer accuracy, while Coder RFT offers broader sampled coverage. Table 1 and Figure 1 show that Coder SFT exceeds RFT in greedy EX (59.06% versus 53.39%) and sampled Pass@1 (58.49% versus 52.27%), but RFT has higher Pass@32 (81.55% versus 77.51%). General SFT likewise leads in greedy EX (55.48% versus 49.35%) and Pass@1 (54.61% versus 47.67%), whereas both paths reach 76.86% Pass@32 after rounding. The supervised paths therefore offer different reliability–coverage trade-offs: SFT produces a stronger individual answer in both backbones, whereas the Coder RFT advantage emerges at larger candidate budgets.

3.3 Changes in reliability and coverage after GRPO

We examine whether adding GRPO improves both low-sample reliability and high-sample coverage. Each GRPO checkpoint is compared with its own pre-RL model under the same development protocol. Figure 2 plots the within-Coder changes computed from unrounded scores; the main comparison includes the corresponding General results under identical evaluation conditions. Low-k changes emphasize answer reliability; high-k changes characterize the availability of successful candidates.

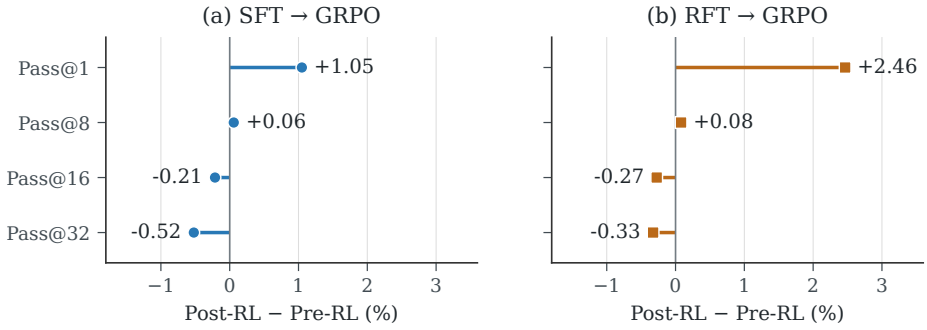

Figure 2: Coder 7B sampled-metric changes after GRPO, relative to each pre-RL model. The zero line distinguishes gains from losses. Values are computed before rounding.

Reliability and coverage respond differently to GRPO. For Coder, GRPO raises greedy EX and Pass@1 for both supervised initializations, while Pass@32 falls from 77.51% to 76.99% after SFT and from 81.55% to 81.23% after RFT. General SFT→GRPO shows the same split: greedy EX rises from 55.48% to 56.65%, but Pass@32 falls from 76.86% to 75.95%. General RFT→GRPO improves greedy EX to 52.09% and Pass@1 to 51.56%, while Pass@32 falls to 74.12%. Coder original→GRPO raises greedy EX to 58.47% but reduces Pass@32 from 79.14% to 67.80%; General original→GRPO reaches 53.85% greedy EX and increases Pass@32 from 60.04% to 64.15%. Direct GRPO therefore has opposite coverage effects across the two backbones. Low-sample suc-

cess and full-pool coverage thus measure different effects of GRPO; paired question-level outcomes further characterize the associated turnover.

4 Analysis of Reasoning Behavior

We analyze template use, changes in freely generated reasoning, and the utility of reasoning organizations on held-out answers. Free generation characterizes the models' preferred output forms; scaffold interventions measure responses to prescribed forms. We examine complete reasoning traces alongside opening labels to characterize both familiar formats and freely organized responses. The sequence connects visible formats, the organization of complete traces, and their association with solving the task.

4.1 Template Use in Freely Generated Answers

Each training question has one target per template; we test how this balanced exposure shapes output frequencies. We compare output forms across training and decoding. Fixed first-line rules identify CHESS, DIN-SQL, E-SQL, Plan, Direct (SQL-block openings), and unmatched openings. Figure 3 compares Coder SFT before and after GRPO under greedy and 32-sample decoding. Unmatched openings are retained as a separate surface category. Greedy frequencies count one answer per question; sampled frequencies count all 32 draws, revealing alternatives beyond the dominant format.

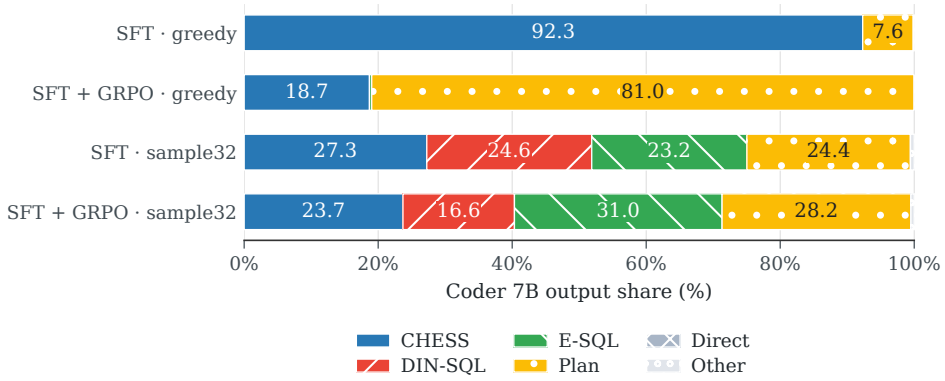

Figure 3: Coder 7B first-line marker frequencies under identical classification rules. Greedy rows contain 1,534 outputs each; sampled rows contain 49,088 each. Labels show shares of at least 7%. Direct denotes SQL-block openings; Other denotes unclassified openings.

Output forms concentrate despite balanced template exposure. Coder greedy outputs shift from 1,416 CHESS and 116 Plan markers under SFT to 287 CHESS and 1,242 Plan after GRPO; General shifts from 1,164 CHESS and 365 Plan to 87 CHESS and 1,411 Plan. Both concentrate toward Plan, while sampled Coder outputs remain distributed across all four templates. The two decoding modes therefore exhibit distinct format distributions. Coder RFT→GRPO produces unmatched openings under the same four-marker classification rule. Full-prose function profiles provide a separate organization test in Section 4.3. These distributions show concentrated format reuse and changing preferences despite balanced exposure to the demonstrations. Same-question marker contrasts include zero in 18 of 24 exploratory intervals (Appendix D.2).

4.2 Effects of Prescribed Reasoning Templates

We test responsiveness to external instructions by varying the requested scaffold for each fixed question. Eight supervised/GRPO models answer 256 questions selected independently of answer correctness under neutral, CHESS, DIN-SQL, E-SQL, and Plan instructions. Conditions share the question and schema, greedy decoding, a 4,096-token output cap, and a 16,384-token context limit.

The $8 \times 5 \times 256$ model–instruction–question combinations yield 10,240 answers. Figure 4 gives matched contrasts; Appendix E reports accuracies and failures for the scaffold and prefix interventions. Neutral prompting supplies the reference for each paired contrast. Testing every scaffold with every model compares instruction effects across model families and training methods. This complements free generation by testing responses to explicitly supplied forms, with failures retained in all conditions.

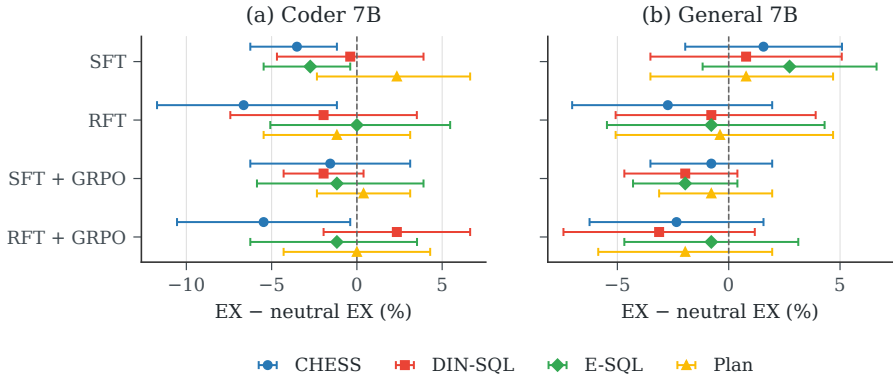

Figure 4: Paired accuracy changes under requested templates on 256 questions. Error bars show unadjusted 95% intervals from 2,000 bootstrap samples of questions. Each contrast retains all questions, including failed or truncated answers.

Observed scaffold effects vary across templates and model states. Changes range from -6.64% to $+2.73\%$; 28 of 32 intervals include zero. CHESS lowers Coder RFT from 50.00% to 43.36%. Coder SFT has an observed accuracy of 58.59% with Plan versus 56.25% under the neutral prompt; the difference is 0.39% after GRPO. General SFT has its largest positive point estimate under E-SQL (+2.73%). Both positive SFT contrasts have intervals including zero. The prompts with the highest observed accuracy differ between the two SFT models, contrasting with their shared shift toward Plan markers in neutral greedy outputs.

4.3 Changes in Reasoning Organization

We test whether freely generated reasoning changes in its functional organization rather than only its opening labels. Fixed, schema-grounded prose rules detect eight expressed functions: schema correspondence, decomposition, filtering, joining, aggregation, ordering, nested-query planning, and checking. For each response, we form an ordered function profile by following the first occurrence of each function in the reasoning prose; code and headings are excluded. The same extractor processes 65,088 training targets and 404,976 greedy and sampled answers from eight supervised/GRPO states. Table 2 compares greedy answers to the same questions, retaining pairs with observable profiles in both states. Appendix F details profile extraction, query alignment, and paired correctness analyses. The comparison separates changes in which functions are expressed from changes in their order. Added and removed functions describe profile membership; reordering examines only functions shared by both answers. Pairing responses to the same question holds the task fixed while comparing these aspects of organization.

Table 2: Functional organization changes after GRPO. Paired counts are out of 1,534 questions per path. Added/removed count pairs with at least one detected function added/removed and can overlap. Reordered counts pairs whose common functions change relative order, requiring at least two common functions.

Model	Method	Paired	Added	Removed	Reordered
Coder	SFT	1482	1008	1098	131
Coder	RFT	1519	450	458	350
General	SFT	1491	934	1027	139
General	RFT	1400	518	459	234

Free CoT changes beyond opening markers. Profiles change on 1,167/1,482 and 885/1,519 observable Coder SFT/RFT pairs, and 1,090/1,491 and 857/1,400 General pairs. SFT paths show more widespread function additions and removals, whereas RFT paths show more reordering of common functions. These overlapping counts establish how often detected function membership or order changes across paired responses. They characterize the organizational part of the transition under a common extraction rule. Section 4.4 then examines its task value: whether organizations associated with successful solutions become more frequent after RL and retain an accuracy advantage on held-out candidates.

4.4 Frequency and Utility of Reasoning Organizations

We test the utility of these full-prose profiles on 392,704 sampled answers. Each question's 32 candidates are split into even/odd halves; we select a success-associated profile in one half and evaluate on the other, exchanging roles. Selection maximizes smoothed accuracy among at least two profiles occurring four times each. Pooled selection uses pre-RL and post-RL fitting halves; pre-RL selection excludes GRPO outcomes. Held-out correctness comparisons match within question and a 1.2 token-length ratio, requiring two pairs in both states. Appendix G gives model- and question-transfer controls. The pre-RL comparison asks whether GRPO makes previously useful organizations more common, with usefulness identified entirely before reinforcement learning. Separate selection and evaluation halves, paired questions, and length matching connect expression frequency with execution quality at comparable response budgets.

Table 3: Frequency shifts (Δf, %) and held-out GRPO accuracy gaps. N_f/N_m are frequency/matched question counts under pre-RL selection out of 1,534; cohorts differ. Frequency denominators include all 16 held-out candidates. Brackets give 95% intervals from 2,000 bootstrap samples of questions.

Model	Method	Pooled Δf	Pre-RL Δf	N_f/N_m	EX gap
Coder	SFT	+2.04	+1.80 [+1.28, +2.29]	1041/627	+3.76 [+1.93, +5.60]
Coder	RFT	+2.19	+2.22 [+1.01, +3.60]	342/255	+1.06 [-1.10, +3.09]
General	SFT	+3.29	+2.87 [+2.20, +3.49]	970/587	+6.20 [+4.11, +8.16]
General	RFT	+8.65	+4.00 [+1.87, +6.05]	327/194	-0.02 [-2.80, +2.79]

Pre-RL selection retains positive accuracy gaps for SFT. Pooled selection yields positive held-out gaps on all four paths (1.61–8.97%). All frequency shifts remain significant after Holm correction. Removing GRPO outcomes from selection preserves every frequency shift, but positive length-matched accuracy gaps remain for SFT (Table 3). The selected profiles already occur before RL and become more frequent in held-out post-RL responses. Own-model accuracy differences vary across training states, and ordered cross-question contrasts have intervals including zero (Appendices G.3 and G.4). Together with functional changes, the SFT results support behavioral recombination as increased use of success-associated free CoT organizations. These associations compare held-out responses to the same question after profiles are selected on separate fitting candidates.

5 Sensitivity and Complementarity

Prefix-content controls, cross-model complementarity, and budget sensitivity characterize how reasoning changes relate to performance. A supplementary 32B comparison examines reliability and coverage at a larger model scale. Appendix B examines correctness turnover and sampling concentration; Appendix D relates output forms and likelihood to execution quality.

5.1 Sensitivity to question–prefix alignment

To measure how the supplied reasoning text affects continuation accuracy, we replace a pre-RL model's own reasoning prefix with one from another question in the same database, or reorder its own heading-delimited blocks. Prefixes come from the first sampled candidate, without correctness

filtering; the terminal SQL is removed. Donors share a 50-word length bin. Pre-RL and post-RL checkpoints receive identical pre-RL prefixes as unfinished assistant turns, and only the new continuation is scored. Table 4 reports all 44 Coder and 41 General SFT-eligible questions under greedy decoding and a 4,096-token continuation cap. Contrasts are paired within question and retain failed continuations in their scoring denominators.

Table 4: Question–prefix alignment for SFT-derived answers meeting the prefix-selection criteria. Correct counts use the same questions within each backbone. Changes are other-question minus own-prefix EX differences (%), with unadjusted 95% paired intervals from 2,000 bootstrap samples of questions. RFT results cover one Coder and six General questions and appear with the complete prefix comparisons.

Model	Method	n	Own	Donor	Reordered	Donor change [95% interval]
Coder	SFT	44	30	10	28	-45.45 [-61.36, -31.82]
Coder	SFT + GRPO	44	30	13	30	-38.64 [-54.55, -22.73]
General	SFT	41	26	2	26	-58.54 [-73.17, -43.90]
General	SFT + GRPO	41	26	5	26	-51.22 [-68.29, -34.15]

Continuations are sensitive to question–prefix alignment. Donor-minus-own EX differences range from -58.54% to -38.64% across the four states, with all exploratory intervals below zero; block reordering changes correct counts by at most two. The accuracy loss quantifies sensitivity to replacing the question-specific reasoning text. The substitution changes both content and organization within the matched length bin, whereas block reordering preserves the original content. This complements the prescribed-template comparison in Section 4.2: continuation accuracy varies with the match between the question and supplied prefix. Donor sensitivity occurs both before and after GRPO; intervals for its post-training reduction include or touch zero (Appendix E.4).

5.2 Cross-backbone complementarity as performance context

We examine whether similar aggregate scores conceal complementary successes. Figure 5 pairs all 1,534 questions, separating greedy correctness from coverage by 32 samples per model. The union measures oracle coverage across both models, using two greedy answers or 64 sampled answers per question.

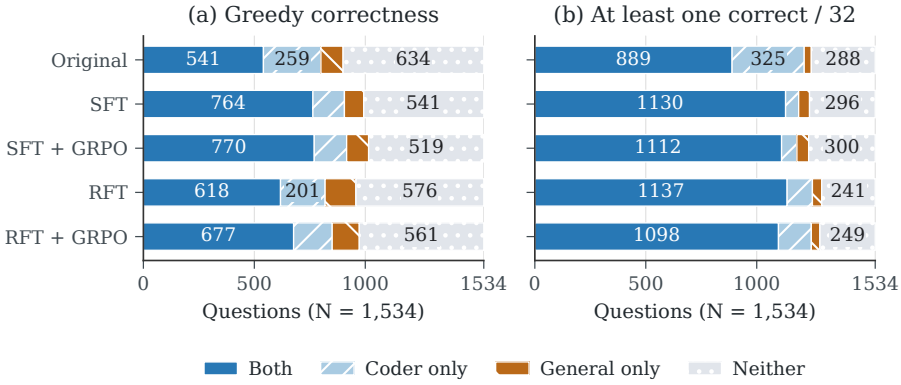

Figure 5: Cross-backbone correctness overlap on the full development set. 32-sample coverage tests whether each model has at least one correct candidate. Small cells are unlabeled; complete counts appear in Appendix B.2.

Complementary successes persist after GRPO. SFT/RFT greedy unions rise from 993/958 to 1,015/973 questions, while their 32-sample unions fall from 1,238/1,293 to 1,234/1,285. Improved greedy reliability therefore coexists with reduced joint coverage. The paired overlaps distinguish persistent shared successes from gains unique to either model. The union counts a question as solved when either model succeeds. For greedy decoding, this combines two answers per question;

for sampled decoding, it combines 64 candidates. Joint coverage thus measures the availability of successful answers across models, complementing each model's accuracy and coverage.

5.3 Sensitivity to the generation budget

We assess how GRPO changes sensitivity to the maximum answer length. Eight checkpoints answer the same 256 hash-selected questions with ordinary prompts, greedy decoding, seed 0, and caps of 512, 2,048, and 4,096 new tokens, yielding $8 \times 3 \times 256 = 6{,}144$ answers. Table 5 reports the accuracy penalty at 512 relative to 2,048 tokens, retaining every failure. Appendix C reports candidate-subset stability, output lengths, failures, and cap-specific accuracies.

Table 5: Short-budget accuracy penalty (%). Reduction is before minus after GRPO; brackets give unadjusted 95% paired intervals from 2,000 bootstrap samples of questions.

Model	Method	Before GRPO	After GRPO	Reduction [95% CI]
Coder	SFT	28.13	12.11	16.02 [11.33, 21.09]
General	SFT	21.88	10.94	10.94 [6.25, 15.63]
Coder	RFT	0.39	0.00	0.39 [0.00, 1.17]
General	RFT	0.00	0.00	0.00 [0.00, 0.00]

SFT continuations show smaller short-budget penalties after GRPO. Their penalties fall from 28.13% to 12.11% and from 21.88% to 10.94%, while RFT penalties remain at or below 0.39%. None of the eight checkpoints gains accuracy when the cap increases from 2,048 to 4,096 tokens on this cohort. The largest effect is therefore the 512-token restriction on SFT-derived answers.

5.4 Reliability and Coverage at Larger Model Scales

The supplementary Qwen2.5-Coder-32B-Instruct study combines execution-filtered training and teacher distillation from reference-SQL-conditioned rationales. Table 6 compares sampled Pass@k before and after RL at four budgets, examining reliability and coverage at a larger model scale.

Table 6: Supplementary 32B sampled Pass@k (percent); differences are computed as after minus before from unrounded values.

	Pass@1	Pass@3	Pass@5	Pass@8
Before RL	59.07	72.03	75.50	78.16
After RL	68.03	73.95	75.86	77.57
Difference	+8.96	+1.92	+0.36	-0.59

Single-sample success improves more than multi-candidate coverage. In this 32B comparison, Pass@1 rises from 59.07% to 68.03%, while Pass@8 falls from 78.16% to 77.57%. As in the 7B comparisons, the largest improvement occurs at the smallest budget. Post-RL greedy EX is 68.57%; it evaluates one deterministic answer per question, whereas sampled Pass@1 estimates the success probability of a single stochastic draw.

6 Conclusion

After multi-template fine-tuning, GRPO changes both template preferences and reasoning organization. First, SFT greedy outputs shift from CHESS toward Plan despite balanced template counts, and prescribed-template effects vary across models. Second, full-prose profiles show additions, removals, and reordering of reasoning functions. Third, held-out candidates increasingly use success-associated organizations. Selecting organizations from pre-RL answers alone preserves this frequency increase on every supervised path, with a positive accuracy advantage for SFT. These findings characterize strategy recombination through the increased use of useful CoT organizations. GRPO improves greedy accuracy while reducing Pass@32 across supervised paths.

AI USE STATEMENT

Generative AI assisted language polishing, manuscript organization, and translation, as well as experiment and analysis design, implementation, data consistency checks, and interpretation of results. Language models also generated the training trajectories described in the experimental protocol. The authors reviewed the AI-assisted text and code, checked numerical claims against experimental outputs, and retain responsibility for the methodology, results, and final manuscript.

ETHICS STATEMENT

This study investigates reasoning behavior in text-to-SQL models using the BIRD benchmark. The experiments evaluate SQL generation for benchmark questions and databases. Generated SQL can be syntactically valid yet return incorrect or sensitive information. Practical use requires access controls, restricted execution permissions, and review appropriate to the consequences of an erroneous query. Query correctness should be assessed through execution and result verification, alongside inspection of the generated reasoning. Use and redistribution of the underlying data and models should follow their respective licenses and access conditions.

REPRODUCIBILITY STATEMENT

The empirical study and experimental-protocol appendix specify the 7B training data, model initializations, optimization, decoding, and execution scoring. The analysis appendices define the compared populations and statistical procedures. The anonymous code supplement provides selected training launchers, execution evaluators, reasoning-profile extraction, and statistical analyses, with dependency versions and interface requirements. Models, BIRD databases, and external training frameworks are obtained separately; the supplement README specifies their input formats and how to run the supplied code.

REFERENCES

Hasan Alp Caferoğlu and Özgür Ulusoy. E-SQL: Direct schema linking via question enrichment in text-to-SQL. *arXiv preprint arXiv:2409.16751*, 2024. URL https://arxiv.org/abs/2409.16751.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Hojae Han, Yeonseok Jeong, Seung-won Hwang, Zhewei Yao, and Yuxiong He. R^3-SQL: Ranking reward and resampling for text-to-SQL. *arXiv preprint arXiv:2604.25325*, 2026. URL https://arxiv.org/abs/2604.25325.

Binyuan Hui, Jian Yang, Zeyu Cui, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024. URL https://arxiv.org/abs/2409.12186.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. *arXiv preprint arXiv:2305.03111*, 2023. URL https://arxiv.org/abs/2305.03111.

Long Li, Zhijian Zhou, Jiaran Hao, et al. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.07430.

Mohammadreza Pourreza and Davood Rafiei. DIN-SQL: Decomposed in-context learning of text-to-SQL with self-correction. *arXiv preprint arXiv:2304.11015*, 2023. URL https://arxiv.org/abs/2304.11015.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Shayan Talaei, Mohammadreza Pourreza, Yu-Chen Chang, Azalia Mirhoseini, and Amin Saberi. CHESS: Contextual harnessing for efficient SQL synthesis. *arXiv preprint arXiv:2405.16755*, 2024. URL https://arxiv.org/abs/2405.16755.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171.

An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Zhewei Yao, Guoheng Sun, Lukasz Borchmann, Zheyu Shen, Minghang Deng, Bohan Zhai, Hao Zhang, Ang Li, and Yuxiong He. Arctic-Text2SQL-R1: Simple rewards, strong reasoning in text-to-SQL. *arXiv preprint arXiv:2505.20315v1*, 2025. URL https://arxiv.org/abs/2505.20315v1.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. URL https://arxiv.org/abs/2504.13837.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*, 2022. URL https://arxiv.org/abs/2203.14465.

A Experimental Protocol and Analysis Definitions

A.1 Models and matched training data

Models and training paths. The two backbones each have six states (original, SFT, RFT, and GRPO from each initialization), giving ten training stages at seed 0. SFT and RFT start independently from original weights; direct GRPO starts from the original instruction-tuned model.

Matched training data. The training cohort contains 5,424 BIRD training questions shared by three answer sources: teacher trajectories generated by Qwen2.5-Coder-32B-Instruct and trajectories generated by each original 7B backbone. Each source contains one execution-correct answer per question under each of CHESS, DIN-SQL, E-SQL, and Plan, totaling 21,696 trajectories. Both SFT runs use the same teacher targets; each RFT run uses its own backbone's generated targets. Supervision uses the full matched training cohort. All 1,534 development questions remain separate from training, rejection filtering, and checkpoint selection.

The initial common intersection contains 5,427 questions. Three are jointly excluded after SQL verification, including a numerical-division mismatch and two unresolved execution cases. Exclusions apply to every source and template. Answers that time out initially but pass extended execution verification remain for retained questions; initial and extended execution outcomes are distinguished. This construction matches template coverage on questions answerable under all four templates by all three sources.

RFT data construction and exposure. Initial generation covers 7,238 questions, four structural instructions, and eight candidates per question–template pair for each original model. Further sampling targets missing pairs within the 5,954-question teacher pool having all four templates and stops after twenty additional rounds. Each round generates eight candidates per remaining pair at temperature 0.8, top-p 0.95, and at most 2,048 new tokens; successful pairs are not repeatedly sampled. The final training set is the matched intersection above. The guides provide neither examples nor reference SQL. Each retained target is the first successful, nontruncated candidate in sampling order. Training removes the generation-only guide and uses the ordinary question prompt. GRPO receives one ordinary prompt per training question and inherits template exposure from its own pre-RL model. Ordinary CoT and direct SQL are classified as output behaviors; supervision uses the four named template sources.

A.2 Supervised target-token exposure

We test whether matching questions, template counts, epochs, and updates also matches target-token exposure. Table 7 separates total from loss-bearing target tokens, counted with the training tokenizer. Counts exclude padding and target-generation cost; the training schedule is identical across supervised runs.

Table 7: Tokens per epoch, counted with the actual tokenizer. Each row has 21,696 answers; supervised tokens bear loss. Three epochs triple these counts. Padding and FLOPs are excluded.

Model	Method	Total tokens	Supervised tokens
Coder 7B	SFT	56,337,482	8,386,313
Coder 7B	RFT	56,151,731	8,200,563
General 7B	SFT	56,337,482	8,386,313
General 7B	RFT	54,600,712	6,649,544

Matched template counts yield different token exposure. Both backbones share the SFT teacher targets, but RFT uses 2.21% fewer supervised tokens for Coder and 20.71% fewer for General. Matching question and template coverage therefore leaves source-specific differences in target content and token exposure.

A.3 Optimization, evaluation, and statistical analysis

Optimization. Section 3.1 specifies schedules and batch sizes. SFT and RFT use cosine scheduling, warm-up 0.03, 16,384-token context, BF16, gradient checkpointing, and SDPA; overlap communication is disabled. Each supervised run performs 1,017 optimizer updates. OpenRLHF GRPO uses group-normalized advantages, a clipped PPO surrogate, sequence-normalized loss, and k_3 KL coefficient 0.01. Training batch is 128, with one optimization epoch per rollout; dynamic filtering, asynchronous updates, and extra importance-sampling correction are disabled. GRPO adds a stage of optimization using execution feedback.

Execution reward and decoding. Binary reward compares the final SQL block with the reference execution result; the reward depends solely on execution correctness. Missing databases and invalid reference queries are classified separately from model execution errors. Prediction timeouts are 30 seconds in training and 600 seconds in development. One slow reference is cached by exact SQL/database hash. Evaluation uses normalized chat prompts, a 32,768-token context that retains complete prompts, and the common decoding settings and Pass@k definition used in the main evaluation. Failures and truncations remain in all denominators.

Response classification. Prefix-based style counts describe visible response formats (Li et al., 2026). Fixed first-line rules count CHESS, DIN-SQL, E-SQL, Plan, Direct, and unmatched openings separately from training-source membership. The categories describe response openings: Direct identifies SQL-block openings, which may precede further prose, and unmatched responses remain a separate category. Full-prose function profiles provide the complementary organization measure.

Execution and reporting. Supervision uses eight-GPU ZeRO-1 (per-device batch 1, accumulation 8); GRPO uses eight-GPU ZeRO-3 with eight colocated single-GPU rollout engines. Generation uses eight GPUs in parallel; SQL execution is evaluated with eight CPU processes, checked against 264 serially scored responses with identical outcomes. Evaluation outputs retain question and candidate identities, model identity, execution scores, token counts, and software and data versions.

Baseline evaluation consistency. Original-model evaluations use identical weights, question sets, execution evaluators, database versions, tokenization, and decoding settings. Each answer is evaluated for execution correctness, failure type, and generated token count.

Schema representation. All checkpoints use the same schema serialization for the 1,534 evaluation prompts. The supplied schemas include a `Cudói` table absent from the execution database in 145 toxicology prompts (21 in the 256-question analysis cohort). Every model receives the same supplied schema for each question.

Analysis populations. The controlled comparisons use the common training intersection and predetermined final checkpoints, evaluated on BIRD development data. The supplementary Qwen2.5-Coder-32B-Instruct comparison evaluates pre-RL checkpoint 120 and its RL continuation on 1,534 BIRD development questions. Stochastic decoding uses temperature 1.0, with Pass@k reported at $k = 1, 3, 5, 8$; post-RL greedy accuracy is evaluated separately. Its execution-filtered training and reference-SQL-conditioned teacher distillation form a separate training protocol from the matched 7B study.

The analyses cover paired question-level gains (Section B.1), cross-backbone overlap (Section 5.2), fixed-candidate likelihood (Section D.3), and the supplementary prefix, sampling, length, and format analyses. Overlap, length, and failure analyses use all 24 evaluation conditions; repetition and coverage partitioning use the 12 sampled conditions, and format–quality contrasts use the four SFT-derived sampled conditions. Likelihood analyses use the checkpoint and candidate populations specified in each analysis. Likelihood scoring uses a fixed, correctness-independent 256-question sample and fixed candidate texts.

The inferential unit is the question, with candidates grouped within questions. Same-question comparisons estimate associations among the generated forms, lengths, and execution outcomes.

Bootstrap intervals resample questions and use no multiple-comparison correction unless explicitly stated. Fixed-candidate likelihood scores the same texts under different checkpoints.

A.4 Training rollout and time budgets

Table 8 reports generated action tokens and summed phase durations across 339 rollouts per GRPO branch, separately from supervised target-token exposure.

Table 8: GRPO rollout budgets. Every row uses 5,424 prompts, 339 rollouts, and 43,392 candidates. Action-token totals exclude prompt tokens and training forward/backward computation. Phase time sums candidate generation, training-batch construction, policy optimization, and weight synchronization across the 339 rollouts.

Model	Method	Action tokens	Truncated	Phase time (h)
Coder	Original + GRPO	2,330,155	0	4.40
Coder	SFT + GRPO	16,930,090	12	4.65
Coder	RFT + GRPO	16,077,037	5	5.07
General	Original + GRPO	2,420,409	1	4.40
General	SFT + GRPO	16,529,232	16	4.69
General	RFT + GRPO	12,299,457	3	4.64

Supervised initializations generate more action tokens. Equal prompt and candidate counts yield 2.3–2.4 million action tokens per backbone for direct GRPO, versus 12.3–16.9 million for supervised continuations. Supervised training takes 4,977.6/4,996.2 seconds for Coder SFT/RFT and 4,987.8/4,940.4 seconds for General. GRPO phase totals range from 4.40 to 5.07 hours, including 4.69/4.64 hours for General SFT/RFT. Action-token counts exclude prompt processing, optimizer computation, and SQL execution.

B Correctness and Sampling Concentration

B.1 Correctness transitions after GRPO

We pair pre-RL and post-RL greedy execution labels on all 1,534 questions to test whether accuracy gains preserve previously solved questions. Figure 6 separates gains from regressions; their difference gives the net accuracy change in question counts. Comparisons use fixed final checkpoints and retain unchanged outcomes.

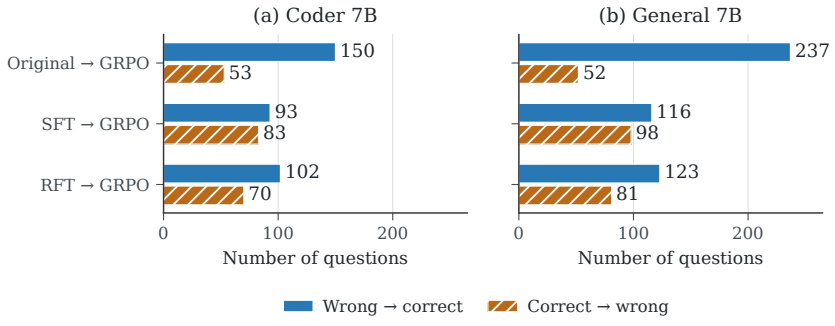

Figure 6: Greedy correctness transitions after GRPO on all 1,534 questions. Solid bars show gains and hatched bars regressions. Each comparison uses its own pre-RL model; counts represent paired outcomes on the same questions.

Accuracy gains coexist with losses on previously solved questions. All six paths gain more questions than they lose, with direct GRPO adding 97/185 correct answers for Coder/General. Coder's

falling Pass@32 accompanies these greedy gains, separating single-answer improvement from sampled coverage. Length changes also depend on decoding (Appendix C.2).

B.2 Cross-model correctness overlap

Table 9 pairs backbone outcomes on the same questions and reports all disjoint overlap cells. Sampled success requires at least one correct answer among 32 candidates per model. The oracle union thus covers two greedy answers or 64 sampled candidates per question.

Table 9: Cross-backbone overlap counts; every row sums to 1,534 across the four disjoint cells. Union equals 1,534 minus neither.

Method	Decoding	Both	Coder only	General only	Neither	Union
Original	Greedy	541	259	100	634	900
Original	Sample32	889	325	32	288	1246
SFT	Greedy	764	142	87	541	993
SFT	Sample32	1130	59	49	296	1238
RFT	Greedy	618	201	139	576	958
RFT	Sample32	1137	114	42	241	1293
SFT + GRPO	Greedy	770	146	99	519	1015
SFT + GRPO	Sample32	1112	69	53	300	1234
RFT + GRPO	Greedy	677	174	122	561	973
RFT + GRPO	Sample32	1098	148	39	249	1285
Original + GRPO	Greedy	723	174	103	534	1000
Original + GRPO	Sample32	880	160	104	390	1144

Both models contribute exclusive successes. General contributes unique correct answers despite its lower aggregate accuracy. Newly covered question sets intersect on zero questions after SFT→GRPO (15 Coder gains; 19 General) and four after RFT→GRPO (34; 29). For greedy answers, the gain sets intersect on 27 questions after SFT (93; 116) and 25 after RFT (102; 123). Each gain set is defined relative to its own backbone's pre-GRPO answers.

B.3 Sampling concentration

Table 10 compares SQL concentration across 32-candidate sets (1,534 questions; 49,088 answers per state). SQLGlot 27.29.0 parses the last SQL block as SQLite; token-type/text sequences preserve case and literals but ignore whitespace. Distinct counts average over all questions; repetition uses parseable pools, available for all 1,534 questions. Sparse-success shares condition on coverage, and parse failures are counted separately.

Table 10: 32-sample SQL concentration on all 1,534 questions per model state (49,088 candidates). U: mean distinct parseable SQL sequences; R: mean repetition among parseable pools; F: extraction/parse failures. Sparse: covered questions with 1–4 correct candidates.

Model	Method	U	R (%)	F	Sparse / covered
Coder	Original	17.29	45.00	750	185/1214
Coder	SFT	7.09	77.57	197	99/1189
Coder	SFT + GRPO	6.64	79.06	156	93/1181
Coder	RFT	20.30	35.87	408	173/1251
Coder	RFT + GRPO	18.56	41.41	338	162/1246
Coder	Original + GRPO	5.18	83.75	89	51/1040
General	Original	10.58	65.98	740	124/921
General	SFT	8.11	74.07	501	126/1179
General	SFT + GRPO	7.29	76.75	452	103/1165
General	RFT	17.78	42.81	951	184/1179
General	RFT + GRPO	11.60	62.72	686	135/1137
General	Original + GRPO	6.16	80.25	397	86/984

Lexical diversity and successful coverage are distinct. RFT has more distinct SQL strings than SFT in both backbones (20.30 versus 7.09 Coder; 17.78 versus 8.11 General). GRPO reduces diversity after both supervised initializations; General RFT falls to 11.60 while repetition rises from 42.81% to 62.72%. All four comparisons also reduce the conditional sparse-success share, a measure affected by both concentration on frequent successes and loss of infrequent successes. Distinct SQL counts and question coverage differ: General SFT and RFT both cover 1,179 questions in count, despite different diversity and potentially different solved questions.

C Coverage Stability and Output Budgets

C.1 Stability across candidate subsets

We split each question's 32 candidates into indices 0–15 and 16–31, identically before and after GRPO. Table 11 reports coverage changes on all 1,534 questions, counting success if a half contains a correct answer. This fixed-subset measure tests candidate-pool sensitivity and differs from the all-combinations Pass@16 estimator.

Table 11: Post-RL minus pre-RL coverage on two fixed 16-candidate subsets (%). All 1,534 questions remain in each denominator.

Model	Method	Indices 0–15	Indices 16–31
Coder	Original	-9.13	-8.67
Coder	SFT	-0.13	-0.52
Coder	RFT	-0.46	+0.07
General	Original	+5.35	+4.76
General	SFT	-0.20	-0.85
General	RFT	-1.89	-3.00

Large coverage changes persist across candidate subsets. The larger changes agree across halves: direct GRPO decreases Coder coverage by 9.13%/8.67% and increases General coverage by 5.35%/4.76%; both General supervised continuations decline. Coder RFT instead changes sign (-0.46% versus $+0.07\%$), indicating sensitivity of its near-zero effect to candidate selection.

C.2 Length changes and evaluation failures

Tables 12–13 retain all answers from twelve model states. Length changes pair 1,534 questions, averaging the 32 sampled responses within each question. Timeouts are execution errors; truncation may overlap any outcome. Token totals count completions only.

Table 12: Greedy output lengths and failures for twelve models (1,534 answers each). Execution errors include timeouts; truncation can overlap any execution outcome.

Model	Method	Tokens	Mean	Exec.err.	No SQL	Trunc.	Timeout
Coder	Original	109,026	71.1	227	0	0	0
Coder	SFT	903,901	589.2	118	2	5	1
Coder	RFT	499,064	325.3	203	0	0	0
Coder	Original + GRPO	78,879	51.4	104	0	0	0
Coder	SFT + GRPO	730,583	476.3	120	0	3	0
Coder	RFT + GRPO	472,199	307.8	174	0	0	0
General	Original	102,082	66.5	362	2	1	3
General	SFT	864,070	563.3	200	1	3	0
General	RFT	431,388	281.2	280	0	0	0
General	Original + GRPO	104,886	68.4	210	0	0	2
General	SFT + GRPO	683,681	445.7	165	1	2	0
General	RFT + GRPO	407,219	265.5	244	0	0	1

Sampled decoding evaluates 32 answers per question, giving 49,088 answers per model state. Its length and failure counts use this denominator; greedy decoding uses 1,534.

Table 13: 32-sample output lengths and failures for twelve models (49,088 answers each). Execution errors include timeouts; truncation can overlap any execution outcome.

Model	Method	Tokens	Mean	Exec.err.	No SQL	Trunc.	Timeout
Coder	Original	7,609,897	155.0	8748	367	15	12
Coder	SFT	20,483,977	417.3	4310	14	113	2
Coder	RFT	19,366,944	394.5	6777	8	14	47
Coder	Original + GRPO	2,538,234	51.7	3651	0	0	6
Coder	SFT + GRPO	20,921,055	426.2	3874	13	171	5
Coder	RFT + GRPO	18,284,459	372.5	5760	7	5	24
General	Original	3,269,258	66.6	11921	17	4	92
General	SFT	20,385,215	415.3	6517	21	102	1
General	RFT	15,962,284	325.2	9677	15	18	36
General	Original + GRPO	3,418,982	69.7	7104	18	7	33
General	SFT + GRPO	20,430,122	416.2	5653	18	235	5
General	RFT + GRPO	13,628,338	277.6	7854	10	12	9

Length changes depend on decoding and training path. After GRPO, greedy SFT/RFT answers shorten by 113.0/17.5 tokens for Coder and 117.6/15.8 for General. Sampled SFT changes are $+8.9/+0.9$ (General's interval includes zero); RFT shortens by 22.1/47.5. Direct-GRPO sampled tokens fall from 7.61M to 2.54M for Coder but rise from 3.27M to 3.42M for General. Across twelve states, $12 \times 1{,}534 \times 33 = 607{,}464$ greedy and sampled answers include 109 truncated responses with missing SQL, three with timeouts, and 41 with correct execution. Greedy missing-SQL reductions of 2/0/0/0 are smaller than accuracy gains of 10/18/32/42 for Coder SFT, General SFT, Coder RFT, and General RFT.

C.3 Generation-budget details

Section 5.3 reports the short-budget comparison. Table 14 retains all 24 condition-level results for the same 256 questions, distinguishing permitted length from actual tokens and cap hits. Output-token counts exclude prompt processing, initialization, and SQL execution.

Table 14: Generation-budget sensitivity on 256 questions per state and cap. EX is execution accuracy (%); Tokens is mean actual output length; Hits counts answers reaching the cap (out of 256), irrespective of execution correctness. One greedy run per condition.

Model	Method	512			2,048			4,096		
		EX	Tokens	Hits	EX	Tokens	Hits	EX	Tokens	Hits
Coder	SFT	28.12	468.7	144	56.25	588.4	0	56.25	588.4	0
Coder	RFT	49.22	324.6	15	49.61	329.0	0	49.61	329.0	0
Coder	SFT + GRPO	45.70	433.3	74	57.81	476.9	1	57.81	484.9	1
Coder	RFT + GRPO	51.17	299.6	10	51.17	302.0	0	51.17	302.0	0
General	SFT	29.69	462.9	133	51.56	556.0	0	51.56	556.0	0
General	RFT	47.66	280.0	4	47.66	282.2	0	47.66	282.2	0
General	SFT + GRPO	43.75	428.8	62	54.69	449.0	0	54.69	449.0	0
General	RFT + GRPO	50.39	265.6	4	50.39	267.4	0	50.39	267.4	0

Longer caps can add tokens without improving accuracy. One Coder SFT + GRPO answer reaches both larger caps, adding tokens without changing aggregate accuracy. RFT-derived outputs are shorter, with accuracy penalties of at most 0.39% under the 512-token cap. The full-development evaluation uses the common 2,048-token budget.

D Output Form and Likelihood Analysis

D.1 Fixed-prefix likelihood and observed format frequency

We rank four fixed prefixes by summed log probability and negative mean token NLL from raw logits, excluding EOS. Table 15 correlates these ranks with 32-sample format frequencies within 256 hash-selected questions, then averages Spearman correlations equally across estimable questions. Frequencies retain unmatched openings in their denominators; constant rank vectors are excluded as undefined.

Table 15: Mean within-question Spearman correlation between fixed-prefix scores and sampled format frequencies on 256 fixed questions. n counts questions with nonconstant ranks. Total: summed log probability; normalized: negative mean token NLL. Brackets give exploratory 95% question-bootstrap intervals.

Model	Method	n	Total	Normalized
Coder	SFT	255	-0.084 [-0.151, -0.020]	+0.107 [+0.034, +0.181]
Coder	SFT + GRPO	256	-0.185 [-0.239, -0.133]	-0.229 [-0.294, -0.167]
General	SFT	254	-0.044 [-0.111, +0.022]	+0.081 [+0.010, +0.153]
General	SFT + GRPO	256	-0.034 [-0.085, +0.017]	-0.553 [-0.605, -0.500]

Normalization changes the association. Mean-token correlations shift from 0.107/0.081 after Coder/General SFT to $-0.229/-0.553$ after GRPO; summed-probability correlations differ. Recognized outputs number 8,165/8,192 before and 8,172/8,192 after GRPO for Coder, versus 8,163/8,192 and 8,181/8,192 for General. The other four states per backbone contain no recognized markers, leaving all correlations undefined. Frequencies use temperature 0.8 and top-p 0.95; intervals are unadjusted question-bootstrap intervals.

D.2 Within-question output form and execution quality

Figure 7 compares all six pairs of literal template markers within SFT and SFT→GRPO sampled answers from both backbones. For forms a, b, the contrast is $100|Q_{ab}|^{-1}\sum_{q\in Q_{ab}}(\bar{c}_{q,a} - \bar{c}_{q,b})$, where Q_{ab} contains questions with both forms and $\bar{c}_{q,a}$ is mean candidate correctness for form a. Questions receive equal weight; each pair has its own co-occurrence population. Unadjusted intervals use 2,000 bootstrap samples of questions. These comparisons describe associations among jointly generated form, content, and length.

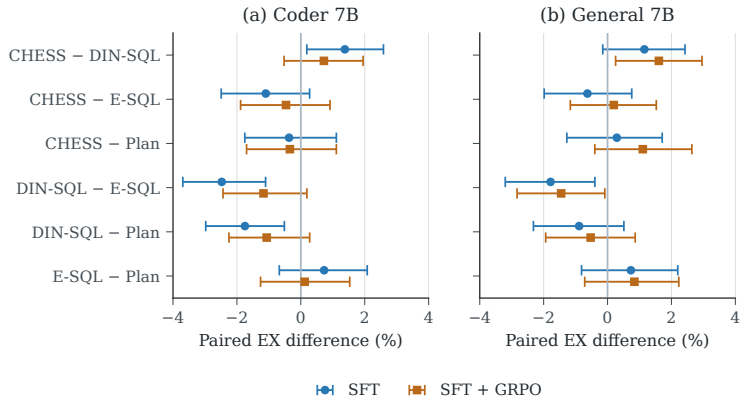

Figure 7: All six within-question marker contrasts for each backbone and stage. Positive values favor the first named form. Bars show exploratory 95% question-bootstrap intervals. Each contrast includes 1,521–1,534 questions with both forms.

Most within-question format contrasts include zero. Mean contrasts range from -2.47% to $+1.61\%$; 18 of 24 intervals include zero. DIN-SQL-minus-E-SQL differences are negative in both SFT backbones (-2.47% Coder; -1.78% General), within the naturally generated candidate pools.

The four markers cover 48,813 and 48,843 of 49,088 Coder responses, and 48,712 and 48,880 General responses before/after GRPO. Mean within-question entropy over all surface categories decreases from 1.935 to 1.900 bits for Coder and 1.934 to 1.854 for General. Original, RFT, and RFT→GRPO lack these four markers under the fixed rule, so the corresponding four-marker quality contrasts are undefined.

D.3 Fixed-candidate likelihood and correctness

We test whether post-training likelihood changes track SQL correctness. A correctness-independent hash selects 256 questions. Each backbone's Original, SFT, and RFT models supply their first sampled answer; duplicate texts are removed. All six states score this fixed pool using completion-token mean NLL, including EOS and masking the prompt. Table 16 averages candidates within questions, then across questions. Correctness gaps require both correct and incorrect candidates; intervals use 2,000 bootstrap samples of questions.

Table 16: Training-induced changes in fixed-candidate mean NLL (after minus before, nats/token) on 256 questions per backbone. Lower NLL means higher likelihood. Brackets give exploratory 95% question-bootstrap intervals (2,000 bootstrap samples of questions).

Comparison	Coder 7B	General 7B
Original → SFT	-0.094 [-0.110, -0.078]	-0.174 [-0.192, -0.157]
Original → RFT	+0.124 [+0.115, +0.133]	+0.098 [+0.085, +0.110]
Original → GRPO	+0.089 [+0.083, +0.094]	+0.048 [+0.041, +0.054]
SFT → GRPO	+0.011 [+0.010, +0.012]	+0.018 [+0.017, +0.019]
RFT → GRPO	+0.005 [+0.004, +0.006]	+0.038 [+0.036, +0.040]

SFT lowers fixed-pool mean NLL by 0.094/0.174 for Coder/General; RFT raises it by 0.124/0.098. Subsequent GRPO yields smaller positive shifts. These changes concern identical candidates scored by different checkpoints. Table 17 instead compares correct and incorrect candidates within the same question and checkpoint.

Table 17: Within-question correct-minus-incorrect mean NLL (nats/token). The comparison includes 80 Coder and 85 General questions containing both correct and incorrect candidates. Negative gaps favor correct candidates. Brackets give exploratory 95% question-bootstrap intervals (2,000 bootstrap samples of questions).

Method	Coder 7B	General 7B
Original	+0.253 [+0.124, +0.372]	+0.394 [+0.242, +0.545]
SFT	-0.202 [-0.295, -0.107]	-0.237 [-0.333, -0.138]
RFT	+0.272 [+0.103, +0.442]	+0.365 [+0.180, +0.561]
Original + GRPO	+0.253 [+0.105, +0.389]	+0.323 [+0.187, +0.458]
SFT + GRPO	-0.211 [-0.308, -0.115]	-0.247 [-0.348, -0.143]
RFT + GRPO	+0.271 [+0.100, +0.441]	+0.364 [+0.172, +0.565]

The likelihood–correctness relationship varies across checkpoints. Correct candidates have lower NLL under SFT and SFT→GRPO, but the gap reverses in other states. These comparisons use a common mixed-source candidate pool, with source, length, and style varying among answers.

D.4 Likelihood-based answer selection

On the same fixed 256-question pools, we compare selection by minimum mean and summed token NLL under all six checkpoints. Each pool contains up to three deduplicated answers: the first sampled answer from each original, SFT, and RFT model. Candidate identifiers break ties in lex-

icographic order. For answer y with T_y completion tokens, mean NLL is $-T_y^{-1}\sum_{t=1}^{T_y}\log p(y_t \mid x, y_{<t})$; summed NLL omits T_y^{-1}. Both include EOS. Table 18 compares Coder selection accuracy against uniform selection and the any-correct oracle.

Table 18: Coder answer selection accuracy (%) on 256 questions. Uniform selection has expected EX of 49.61%; the any-correct oracle accuracy is 64.45%.

Model	Method	Mean-NLL EX	Sum-NLL EX
Coder	Original	44.14	43.75
Coder	SFT	55.08	55.08
Coder	RFT	50.00	48.05
Coder	Original + GRPO	46.88	43.75
Coder	SFT + GRPO	55.08	55.08
Coder	RFT + GRPO	50.00	48.44

Table 19 applies the same selection rules to General. Its uniform and oracle benchmarks are computed from General's own candidate pool.

Table 19: General answer selection accuracy (%) on 256 questions. Uniform selection has expected EX of 45.57%; the any-correct oracle accuracy is 60.94%.

Model	Method	Mean-NLL EX	Sum-NLL EX
General	Original	39.45	39.45
General	SFT	53.12	53.12
General	RFT	44.14	44.92
General	Original + GRPO	39.45	39.45
General	SFT + GRPO	53.12	53.12
General	RFT + GRPO	44.14	45.31

Selection accuracy depends on the scoring rule and model state. Supervised GRPO leaves mean-NLL selection accuracy unchanged in both backbones. Direct GRPO improves Coder selection under mean NLL, while summed-NLL selection remains unchanged. All tested selectors remain below their pool-specific oracle accuracies.

E Template and Prefix Interventions

E.1 Requested-scaffold effects

Table 20 compares five instructions on 256 questions fixed before observing outcomes. Prompts consist of the question, schema, and template instruction; failures remain in all denominators. Paired scaffold-minus-neutral effects are also compared before and after GRPO, distinguishing requested-scaffold interventions from observed output-form associations.

Table 20: Template intervention accuracy (percent) on the same 256 questions. All failures remain in each denominator.

Model	Method	Neutral	CHESS	DIN-SQL	E-SQL	Plan
Coder	SFT	56.25	52.73	55.86	53.52	58.59
Coder	RFT	50.00	43.36	48.05	50.00	48.83
Coder	SFT + GRPO	57.81	56.25	55.86	56.64	58.20
Coder	RFT + GRPO	51.56	46.09	53.91	50.39	51.56
General	SFT	51.56	53.12	52.34	54.30	52.34
General	RFT	48.05	45.31	47.27	47.27	47.66
General	SFT + GRPO	54.69	53.91	52.73	52.73	53.91
General	RFT + GRPO	49.61	47.27	46.48	48.83	47.66

Most changes in scaffold effects have intervals spanning zero. The $8 \times 5 \times 256 = 10{,}240$ answers comprise 5,280 correct executions, 3,714 incorrect results, and 1,246 execution errors, with ten cap hits and 4,233,381 generated tokens. The 32 scaffold contrasts use paired-question intervals and complete prompts whose wording and length vary; the cap is 4,096 rather than the main evaluation's 2,048 tokens. Of 16 pre/post-GRPO effect comparisons, only General SFT E-SQL excludes zero: its effect shifts from $+2.73\%$ to -1.95%, a -4.69% difference ($[-8.98, -0.39]$, unadjusted).

E.2 Sensitivity to template labels

Table 21 compares canonical, removed, renamed, and mismatched labels while holding scaffold bodies fixed. A question hash assigns the scaffold family independently of correctness. Eight checkpoints answer all 256 questions under each condition with a 4,096-token cap, yielding $8 \times 4 \times 256 = 8{,}192$ answers. Unadjusted paired intervals use 2,000 bootstrap samples of questions.

Table 21: Marker perturbation execution accuracy (%) on the same 256 questions. Structural bodies are fixed across label conditions; execution failures and truncated answers remain in every denominator.

Model	Method	Canonical	Removed	Renamed	Mismatched
Coder	SFT	55.47	56.64	55.86	55.86
Coder	RFT	46.09	48.05	46.88	50.78
Coder	SFT + GRPO	57.03	58.98	57.81	57.03
Coder	RFT + GRPO	48.05	50.39	49.61	50.39
General	SFT	53.12	51.95	53.91	51.95
General	RFT	49.22	46.09	49.22	50.00
General	SFT + GRPO	54.30	53.91	54.69	54.30
General	RFT + GRPO	46.48	48.05	48.05	49.22

Most label contrasts have intervals spanning zero. Of 24 contrasts ranging from -3.13% to $+4.69\%$, 23 intervals include zero. These tests isolate label edits with scaffold bodies and question denominators fixed.

E.3 Content and headings of supplied prefixes

We compare no prefix, the pre-RL model's own prefix, and headings alone. Eligibility requires identifiable heading blocks and a removable terminal SQL block in the first sampled answer on the fixed 256-question cohort, without correctness selection. Both checkpoints receive identical prefixes as unfinished assistant turns, greedy decoding, and a 4,096-token continuation cap; only new text is scored. Table 22 includes 57/4 Coder SFT/RFT and 60/12 General questions, yielding $(57 + 4 + 60 + 12) \times 2 \times 3 = 798$ answers across two training states and three conditions.

Table 22: Correct answers / eligible questions under assistant-prefix interventions. Each row retains the same questions across its three conditions. Pre-RL and post-RL checkpoints share the pre-RL-model-specific cohort; cohorts differ across supervised paths. Each cell reports the correct count and the corresponding eligible cohort size.

Model	Method	No prefix	Own prefix	Headings only
Coder	SFT	40/57	39/57	36/57
Coder	RFT	4/4	2/4	3/4
Coder	SFT + GRPO	39/57	39/57	35/57
Coder	RFT + GRPO	4/4	2/4	3/4
General	SFT	37/60	38/60	30/60
General	RFT	6/12	6/12	6/12
General	SFT + GRPO	38/60	38/60	28/60
General	RFT + GRPO	6/12	6/12	6/12

Supplying the original prefix gives little benefit on the SFT-derived cohorts. Correct counts change by at most one answer relative to no prefix. Headings alone reduce correct counts by four for both Coder states and by seven/ten for General SFT and SFT-to-GRPO. Only the General post-GRPO headings contrast has an exploratory interval strictly below zero (-16.67%, 95% paired question-bootstrap interval $[-30.00, -3.33]$). All four SFT-derived post-RL-minus-pre-RL changes in intervention effects include zero. The RFT comparisons cover four Coder and twelve General questions. Each continuation conditions on the supplied reasoning text after removal of its terminal SQL; input length varies with the prefix condition.

E.4 Question relevance and prefix order

We next compare the pre-RL model's own prefix with a prefix from another question in the same database and with a deterministic permutation of its own heading-delimited blocks. Eligible donors share a 50-word length bin, allowing length differences within each bin; permutations preserve block content and word multiplicities. No condition supplies the removed final SQL. Fixed input transformations precede outcome scoring, and each GRPO checkpoint again uses its pre-RL model's prefixes. The stricter eligibility rules retain 44/1 Coder SFT/RFT and 41/6 General SFT/RFT questions. Across two training states and three conditions, these cohorts yield $(44 + 1 + 41 + 6) \times 2 \times 3 = 552$ answers under the same continuation protocol.

For both RFT and RFT + GRPO, the numbers of correct answers under own, other-question, and reordered prefixes are 1/1, 0/1, and 1/1 for Coder, and 3/6, 0/6, and 3/6 for General, respectively. Each fraction uses the eligible question count as its denominator.

Cross-question content is more disruptive than this block permutation. Relative to own prefixes, donor prefixes reduce Coder SFT/post-GRPO EX by 45.45%/38.64% and General SFT/post-GRPO EX by 58.54%/51.22%; all four exploratory paired intervals lie below zero. The corresponding post-RL-minus-pre-RL changes are $+6.82\%$ ($[-2.27, 15.91]$) and $+7.32\%$ ($[0.00, 17.07]\%$), with both intervals including or touching zero. Reordering changes correct counts by at most two on these SFT-derived cohorts. Donor substitution changes content and length within the matched bin; block permutation preserves the original blocks. All intervals are unadjusted and use 2,000 paired bootstrap samples of questions.

F Free Expression and Functional Organization

F.1 Free expression and full-prose organization

We examine complete generated responses for expressed schema correspondence, decomposition, filtering, joining, aggregation, ordering, nested-query planning, and checking. Fixed phrase rules require a real schema identifier within 120 characters and at least six words in the prose unit. For each response, we order functions by their first occurrence in the reasoning prose, counting repeated mentions once and excluding code, repeated questions, and headings. We apply the same extractor to three sources of 21,696 training targets (65,088 total) and eight checkpoints with 1,534 questions and 33 answers per question (one greedy and 32 sampled; 404,976 total). Truncated, malformed, and unmatched answers remain separately counted; unmatched functions are classified as unobserved. The main-text analysis pairs greedy profiles on the same question; Table 23 counts answers whose ordered profile is absent from the matching training source.

Table 23: Greedy answers whose detected ordered profile is absent from the corresponding training source, before and after GRPO. Counts are out of 1,534 questions per path; unobservable profiles remain separately classified.

Model	Method	Before GRPO	After GRPO
Coder	SFT	101	41
Coder	RFT	55	32
General	SFT	102	18
General	RFT	21	27

Reorganization primarily changes the use of familiar functions. Profiles absent from the matching training source decrease in three paths and increase in General RFT; held-out training questions also yield 2.1–3.1% unseen multi-function profiles. The changes primarily concern the frequency and order of familiar functions. Complementary literal cues show Coder/General SFT switching from decomposition to planning on 1,121/1,042 questions, and General RFT switching 349 planning and 67 decomposition outputs to enrichment. Schema-linking cues match 2,330 of 5,424 teacher DIN-SQL targets; full-prose profiles also capture functions expressed elsewhere in the answer.

F.2 Alignment with Reference-Query Structure

We classify all 1,534 reference SQLs offline by joins (1,140), filtering (1,390), aggregation/grouping (670), ordering/limiting (310), and multiple SELECT nodes (126); categories overlap and all queries parse. References are not supplied to generation. For each function, we compare its mention rate on feature-present versus feature-absent questions, then measure the change in this gap after GRPO. Sampled pools are paired at the question level. Table 24 reports all 40 contrasts; uncertainty uses 2,000 stratified bootstrap samples of questions and random-sign draws, with Holm correction over 40 tests. Reference-query features provide shared structural descriptors across the two backbones.

Table 24: Post-GRPO change in the feature-present minus feature-absent prose-mention gap (%). A star denotes Holm-adjusted $p < 0.05$. Positive values indicate stronger alignment between prose mentions and reference-query features.

Model	Method	Decoding	Join	Filter	Aggr.	Order	Multi-select
Coder	SFT	Greedy	-50.6^*	-32.9^*	-10.3^*	-18.2^*	-13.5
Coder	SFT	Sample32	-1.9^*	-2.9^*	$+1.1^*$	-0.7	-1.4
Coder	RFT	Greedy	$+1.9$	$+8.1$	-0.9	-4.0	-2.2
Coder	RFT	Sample32	$+5.4^*$	$+3.4^*$	$+0.5$	$+3.4^*$	-1.5
General	SFT	Greedy	-46.6^*	-23.9^*	-7.0^*	-18.1^*	-20.4^*
General	SFT	Sample32	-2.0^*	-3.2^*	$+1.3^*$	-1.0	-4.6^*
General	RFT	Greedy	$+5.4$	$+7.4$	$+6.6^*$	$+2.9$	$+0.9$
General	RFT	Sample32	$+11.4^*$	$+8.0^*$	$+4.1^*$	$+7.4^*$	-3.7^*

Alignment depends on training path and decoding. Sampled RFT strengthens join/filter/order alignment by 5.4%/3.4%/3.4% for Coder and 11.4%/8.0%/7.4% for General, whereas both SFT greedy paths show large decreases. Ten positive and fifteen negative changes survive correction. The contrasts quantify prose–query alignment under the same extraction rule.

F.3 Paired utility and sensitivity to response length

We relate greedy profile changes to correctness gains and losses across all 1,534 paired questions, counting unobservable profiles separately. Among pairs observable in both states, we regress correctness change on profile change, log-token change, baseline log tokens and correctness, reference-query features, and database fixed effects. Table 25 reports HC3 intervals for these conditional associations. Response-length bands are below 0.9, within 0.9–1.1, and above 1.1 times pre-RL tokens; complete group counts accompany the data.

Table 25: Changed/stable profiles: paired counts and correctness gains/losses (G/L), with length/covariate-adjusted correctness-change associations. The full cohort also includes 52/15/43/134 unobservable pairs in row order; regression uses observable pairs.

Model	Method	Changed: n, G/L	Stable: n, G/L	Adjusted difference (%) [95% CI]
Coder	SFT	1167: 87/76	315: 3/4	$+1.00\ [-2.56, +4.56]$
Coder	RFT	885: 77/61	634: 24/9	$-0.62\ [-3.68, +2.44]$
General	SFT	1090: 104/92	401: 9/2	$+0.50\ [-3.34, +4.35]$
General	RFT	857: 85/56	543: 30/17	$-0.89\ [-4.64, +2.86]$

Changed profiles accompany both gains and losses. In row order, changed groups yield net gains of 11/16/12/29 correct answers, lose 76/61/92/56 previously solved questions, and shorten by 143.2/24.7/159.0/23.1 tokens on average. All four length- and covariate-adjusted intervals include zero (Holm-adjusted $p = 1$). The split-candidate analysis complements these associations by separating profile selection from evaluation.

G Held-Out Utility and Transfer Controls

G.1 Split-candidate selection of useful profiles

Each question's 32 candidates are split by even/odd index, with fitting and evaluation roles exchanged and results averaged ($8 \times 1{,}534 \times 32 = 392{,}704$ answers). Pooled pre/post-RL fitting halves select the highest Beta(1,1)-smoothed accuracy among at least two profiles occurring four times each; ties break lexically. Held-out halves measure frequency and accuracy relative to other observable profiles. Greedy nearest-log-length matching uses a 1.2 token ratio, no replacement, and two pairs per state. Table 26 uses 2,000 question bootstrap samples on distinct frequency and matched cohorts.

Table 26: Split-selected profile frequency changes after GRPO and length-matched held-out accuracy gaps within GRPO outputs. Brackets are 95% intervals. Counts are unique questions, out of 1,534; frequency denominators retain all 16 held-out candidates. All eight tests pass Holm correction across the twelve primary tests of pooled selection and own-model preference.

Model	Method	N_{freq}	Frequency shift (%)	N_{match}	Held-out EX gap (%)
Coder	SFT	1524	+2.04 [+1.71, +2.37]	984	+6.69 [+5.22, +8.27]
Coder	RFT	1366	+2.19 [+1.73, +2.67]	844	+1.61 [+0.39, +2.87]
General	SFT	1519	+3.29 [+2.87, +3.73]	951	+8.97 [+7.21, +10.77]
General	RFT	1379	+8.65 [+7.83, +9.52]	792	+3.12 [+1.84, +4.43]

Post-GRPO samples more frequently use success-associated organizations. All four frequency shifts and held-out gaps are positive; Holm-adjusted $p \leq 0.038$. Before GRPO, the same matched cohorts already have positive gaps of 8.14%/1.83%/8.85%/3.42%. The frequency increase concerns organizations associated with success before RL. A prespecified unordered-function sensitivity retains positive shifts (1.45–7.34%) and post-GRPO gaps (2.93–9.19%). These gaps compare held-out answers to the same question with matched response lengths.

G.2 Transfer of profiles selected before GRPO

Pre-RL fitting halves select profiles with unchanged eligibility, smoothing, tie-breaking, and matching rules. Exchanged held-out halves estimate frequency shifts and GRPO accuracy gaps with separate question-bootstrap intervals. Holm correction covers four tests each of frequency shifts, GRPO gaps, and cross-question contrasts.

Pre-RL-selected profiles become more frequent, with positive SFT accuracy gaps. All four frequency shifts survive correction ($p \leq 0.009$). SFT gaps are 3.76%/6.20% for Coder/General, while both RFT intervals include zero. Unordered profiles retain positive frequency shifts, with the General RFT gap including zero.

G.3 Accuracy of Model-Preferred Organizations

Each model's fitting half selects its most frequent profile, requiring two occurrences. When Coder and General prefer different profiles, held-out answers compare the two with length matching and two pairs per model. Table 27 gives both accuracy gaps and their symmetric average, combining folds within question. Holm correction covers twelve question-level sign tests across pooled selection and model preference.

Table 27: Own-modal versus other-model-modal profile accuracy gaps (%). "Different" counts questions with differing fitting preferences; "Matched" requires held-out length overlap in both models.

Method	Different	Matched	Coder gap	General gap	Mean own gap [95% CI]
SFT	969	89	-0.11	+2.81	+1.35 [-1.38, +3.93]
SFT + GRPO	853	148	-4.66	-1.62	-3.14 [-5.29, -1.15]
RFT	1353	85	-1.03	+2.78	+0.88 [-3.20, +5.25]
RFT + GRPO	1399	96	+0.82	+0.68	+0.75 [-1.68, +3.40]

Own-profile accuracy differences vary across training states. The eligible overlap cohorts contain 85–148 questions per state. Three intervals include zero; SFT + GRPO is negative (-3.14%; Holm $p = 0.020$). The unordered-profile comparison likewise yields intervals that include zero or favor the other model's preferred profile.

G.4 Cross-Question Transfer of Useful Organizations

Within each pre-RL fitting fold and database, questions are sorted by median candidate length, then identifier, and cyclically assigned the next question's selected profile; singletons are excluded. Assignment uses pre-RL fitting data alone. Where target and assigned profiles differ, held-out answers are length-matched within the target question, with two pairs per state. Table 28 compares ordered profiles with prespecified unordered-function sets.

Table 28: Held-out GRPO accuracy for the target question's pre-RL-selected profile minus a different question's profile. "Set" ignores function order and is exploratory. Intervals and significance tests use the predefined minimum of 20 questions; the 15-question row reports its mean.

Model	Method	Ordered N	Own-question gap (%)	Set N	Set gap (%)
Coder	SFT	66	+3.91 [-0.51, +9.34]	93	+6.09 [+0.54, +12.01]
Coder	RFT	25	-2.00 [-8.00, +4.00]	145	+6.76 [+2.85, +10.89]
General	SFT	75	+3.17 [-2.28, +9.11]	102	+7.39 [+2.30, +12.94]
General	RFT	15	+6.67 (descriptive)	54	+2.22 [-3.02, +8.15]

Question specificity depends on representation and overlap. Ordered profiles cover 15–75 questions, with all estimable intervals including zero. Unordered sets cover 54–145 questions and yield positive associations in three paths. Intervals use 2,000 bootstrap samples of questions, holding fitted selectors and donor assignments fixed.