

Training Reasoning Sub-Skills in LLMs with Synthetic Data

Cormac Cureton

M.Sc. Student

Electrical and Computer Engineering
McGill

Abhijeet Praveen

M.Sc. Student

Electrical and Computer Engineering
McGill

Xiaoyin Chen

Phd Sudent, Project Mentor
Computer Science
Université de Montreal

Abstract

This work investigates the potential of fine-tuning large language models (LLMs) using programmatically generated synthetic data to enhance their reasoning sub-skills. The study focuses on search as a foundational reasoning sub-skill and evaluates its transferability to higher-order reasoning tasks, specifically Sudoku and Zebra puzzles. Using Low-Rank Adaptation (LoRA), we fine-tune LLMs on synthetic search trajectories without increasing inference-time computational costs, addressing the challenge of high-quality reasoning data scarcity. Our experiments employ Partial Accuracy and Strict Accuracy metrics to assess the effectiveness of fine-tuning and highlight task-specific performance variations. Results demonstrate that fine-tuning on synthetic search trajectories offers marginal improvements in zero shot Zebra puzzle performance compared to the base model. Synthetically fine-tuned models don't offer any improvements when tested on Sudoku. As expected, models fine-tuned directly on task-specific datasets consistently outperform search-fine-tuned models, emphasizing the value of task-specific data. This study underscored the difficulty in improving model attributes which generalize across tasks. We release all our code to facilitate future research into scalable, efficient methods for enhancing LLM reasoning capabilities.¹

1 Introduction

In recent years, large-scale language models have grown hugely in popularity, being successfully applied to a wide-range of tasks. One factor which seems to be holding back language models from larger use is their struggle to reason effectively, despite huge knowledge bases, LMs still struggle with reasoning tasks (Valmeekam et al., 2022; Stechly

et al., 2024). Approaches like Chain of Thought (Wei et al., 2023), self-consistency (Wang et al., 2023), and step by step verifiers (Lightman et al., 2023) have all been explored to improve performance on reasoning tasks. These approaches improve performance but they also heavily increase the computational costs at inference time.

Instead, our work looks to fine-tune language models for improved performance on reasoning tasks with a single inference pass. By using Low-Rank Adaptation (LoRA), transformer-based language models can be adapted without increasing computational costs at inference-time (Hu et al., 2021).

However, a challenge with improving reasoning in language models is that high-quality reasoning traces are difficult to find and expensive to produce (Bansal et al., 2024). The lack of high-quality reasoning data motivates the desire to fine-tune models with synthetic data. Unfortunately, generating reasoning traces for training is a non-trivial task and using LMs to create traces comes with significant computational cost (Bansal et al., 2024).

A more computationally efficient method to create synthetic data is programmatic generation. This approach has been has been explored to improve LM performance on skills like search and logic (Gandhi et al., 2024; Lehnert et al., 2024; Pi et al., 2022). We hypothesize that these lower-level skills are necessary for performance on higher-level reasoning tasks. Therefore this work will look to fine-tune a base model on synthetic data to improve a lower-level skill, search. We will then assess whether the gains from fine-tuning on synthetic data will transfer to higher-level reasoning tasks. We select Zebra puzzles and Sudoku as two reasoning tasks for this work, more details about the datasets and evaluation can be found in section 4. This work seeks to provide a path to improve

¹<https://github.com/Cormac-C/llm-reasoning-decomp>

78 model’s reasoning ability through fine-tuning on
79 synthetic data for lower-level skills which can be
80 generated at relatively low cost.

81 Our main contributions are:

- 82 1. We open source code which can be used to
83 fine-tune LoRA adapters on reasoning sub-
84 skills with programmatically generated data
85 and measure performance on downstream
86 tasks.
- 87 2. We measure the effect that fine-tuning on
88 search trajectories has on performance on
89 higher-level reasoning tasks.

90 2 Related Work

91 2.1 Pre-training for Reasoning in Language 92 Models

93 Pre-training strategies for enhancing reasoning ca-
94 pabilities in language models have been extensively
95 explored, leveraging various auxiliary tasks and ex-
96 ternal knowledge sources. Notably, ReasonBERT
97 (Deng et al., 2021) integrates knowledge graphs
98 into the pre-training process, enabling the model
99 to establish relationships between entities and im-
100 prove inference. Similarly, LinkBERT (Yasunaga
101 et al., 2022) incorporates hyperlink structures to
102 uncover semantic relationships, demonstrating im-
103 proved reasoning in tasks requiring the synthesis of
104 linked information. These methods illustrate how
105 external data sources can augment reasoning but
106 are often computationally expensive and dependent
107 on specific pre-training corpora.

108 MERit (Jiao et al., 2022) advances reasoning
109 through multi-modal self-supervised learning, com-
110 bining textual and programmatic reasoning chains.
111 While effective, this approach requires vast datasets
112 and heavy computational resources, limiting scal-
113 ability. Such reliance on large-scale pre-training
114 highlights a gap in methods designed for special-
115 ized reasoning tasks where high-quality data is
116 scarce.

117 Our work diverges from these approaches by
118 focusing on fine-tuning rather than pre-training.
119 This shift avoids the resource-intensive nature of
120 pre-training while targeting reasoning sub-skills
121 foundational to broader reasoning tasks. By fine-
122 tuning on programmatically generated data, we
123 embed these sub-skills efficiently, enabling transfer
124 to higher-level reasoning challenges.

2.2 Fine-tuning Models on Synthetic Data

Fine-tuning language models using synthetic data
has gained prominence for its adaptability to task-
specific requirements. Singh et al. (2024) demon-
strated the utility of model-generated synthetic
datasets for enhancing problem-solving capabili-
ties, achieving state-of-the-art performance on rea-
soning benchmarks. Similarly, Bansal et al. (2024)
utilized smaller language models to generate fine-
tuning data for larger models, reducing compu-
tational costs while retaining performance. This
strategy exemplifies the potential of synthetic data
to mitigate resource constraints.

The V-Star framework (Hosseini et al., 2024) em-
ploys a model-based verifier trained on solutions
from other models, enabling error correction and
improved reasoning. Yu et al. (2024) extend this
idea by distilling the complex reasoning of "Sys-
tem 2" models into simpler "System 1" models, al-
lowing efficient fine-tuning for tasks like question
answering. Furthermore, Ye et al. (2024) augment
fine-tuning with both correct and incorrect solu-
tions from the GSM8K dataset, improving error
detection and reasoning accuracy.

A distinct line of work explores training on
search trajectories. Gandhi et al. (2024) fine-tune
models using programmatically generated search
logs for text-based games, emphasizing the devel-
opment of search and logic sub-skills. Similarly,
Lehnert et al. (2024) leverage A* search trajec-
tories for planning tasks. While these approaches
showcase the effectiveness of trajectory-based data
for search and planning, our work expands their
scope as we emphasize the transferability of these
sub-skills to higher-level reasoning tasks.

2.3 Programmatic Data Generation for Reasoning Tasks

Programmatic data generation has emerged as a
cost-effective solution for creating high-quality
datasets tailored to reasoning tasks. This approach
mitigates the challenges of manual annotation, in-
cluding time and expense, by algorithmically gen-
erating reasoning traces.

Pi et al. (2022) introduced a method for gener-
ating reasoning tasks based on program execu-
tion, bridging logical reasoning and programming
paradigms. Shah et al. (2024) demonstrated that
causal transformers trained on programmatically
generated puzzles exhibit enhanced logical reason-
ing, further validating the utility of synthetic data

175 for structured reasoning tasks.
176 Recent work by [Gandhi et al. \(2024\)](#) and [Lehnert et al. \(2024\)](#) applied programmatic generation
177 to search-based tasks, producing datasets tailored
178 to improve model performance on planning and
179 search problems. While these studies focus on spe-
180 cific sub-skills, our approach extends this paradigm
181 by targeting a broader spectrum of reasoning abil-
182 ities. By generating synthetic data for sub-skills
183 such as search, logic, and arithmetic reasoning, we
184 enable fine-tuning that is both computationally effi-
185 cient and effective across diverse reasoning tasks.
186

187 **2.4 Comparison to Prior Work**

188 Compared to existing works in synthetic data gener-
189 ation, particularly for mathematical reasoning, our
190 approach introduces several distinct advantages.
191 Previous methods, such as those by [Ye et al. \(2024\)](#)
192 and [Pi et al. \(2022\)](#), test their models on a single
193 domain like arithmetic or program execution. In
194 contrast, our work attempts to generalize applica-
195 tion of the search sub-skill, testing on different
196 types of reasoning puzzles.

197 Moreover, unlike [Bansal et al. \(2024\)](#), who rely
198 on language models to generate reasoning traces,
199 we adopt a programmatic generation strategy, a
200 more computationally efficient approach. This dis-
201 tinction opens the door for cheaper scaling, using
202 simpler datasets which can be generated at a frac-
203 tion of the computational cost.

204 Finally, [Gandhi et al. \(2024\)](#) showed improve-
205 ments on search tasks from training on search tra-
206 jectories. In our work we look to measure whether
207 those gains can generalize to broader reasoning
208 problems, namely Sudoku and Zebra puzzles.

209 Our work represents a step forward in leveraging
210 programmatic data generation to enhance reason-
211 ing capabilities in language models. We look to
212 provide a scalable approach to model fine-tuning
213 for improved performance on diverse reasoning
214 tasks.

215 **3 Models and Approach**

216 This work looks to test whether fine-tuning on
217 search trajectories will show positive transfer to
218 the selected reasoning tasks, Sudoku and Zebra
219 Puzzles. The base models for all models tested
220 come from the Llama 3.2 family of models², all

experiments were conducted with the 1B and 3B
221 parameter models.
222

Due to computational constraints, all fine-tuning
223 within this work used LoRA adapters ([Hu et al.,](#)
224 [2021](#)). In future, testing the same research direc-
225 tion with full fine-tuning would be an interesting
226 direction for continued work.
227

228 **3.1 Baselines**

To evaluate the effect of fine-tuning on synthetic
229 search trajectories, we compare against two base-
230 lines: base models and models fine-tuned on rea-
231 soning puzzles. If the hypothesis is true, we would
232 expect the search fine-tuned models to outperform
233 the base models and approach the performance of
234 the models directly fine-tuned on the puzzles. The
235 base model is a necessary baseline to check that
236 the search fine-tuning is offering improvement over
237 the model’s abilities. Synthetic fine-tuning is not
238 expected to match the performance of the puzzle-
239 fine-tuned models. We hope to see improvements
240 the synthetic fine-tuning so that performance falls
241 between the two baselines.
242

243 **3.2 Proposed Approach**

The proposed models are fine-tuned on the syn-
244 thetic countdown search trajectories, then perfor-
245 mance is measured on each of the two puzzles,
246 Sudoku and Zebra puzzles. The trajectories are
247 generated using the method introduced in [Gandhi](#)
248 [et al. \(2024\)](#). For each trajectory, the countdown
249 task is introduced in a user message and the full
250 search trajectory and solution are formatted into
251 the assistant’s response. Loss is only calculated on
252 the assistant’s message, focusing on the model’s
253 ability to create the trajectory and reach a solution.
254 Further details on how each instance is formatted
255 is included in Appendix A.
256

For each base model, we test performance after
257 training with two quantities of synthetic data, first
258 with 1,000 samples and then increasing to 10,000
259 samples. Both of the trained models are evaluated
260 in both zero-shot and few-shot contexts ($k=3$), all
261 results are available in Section 6.
262

263 **4 Datasets and Evaluation Metrics**

264 **4.1 Datasets**

To evaluate the effectiveness of our approach,
265 we tested model performance on two datasets of
266 reasoning puzzles: **Sudoku** and **Zebra**. These
267 datasets were selected to test the model’s ability to
268

²[https://www.llama.com/docs/
model-cards-and-prompt-formats/llama3_2/](https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/)

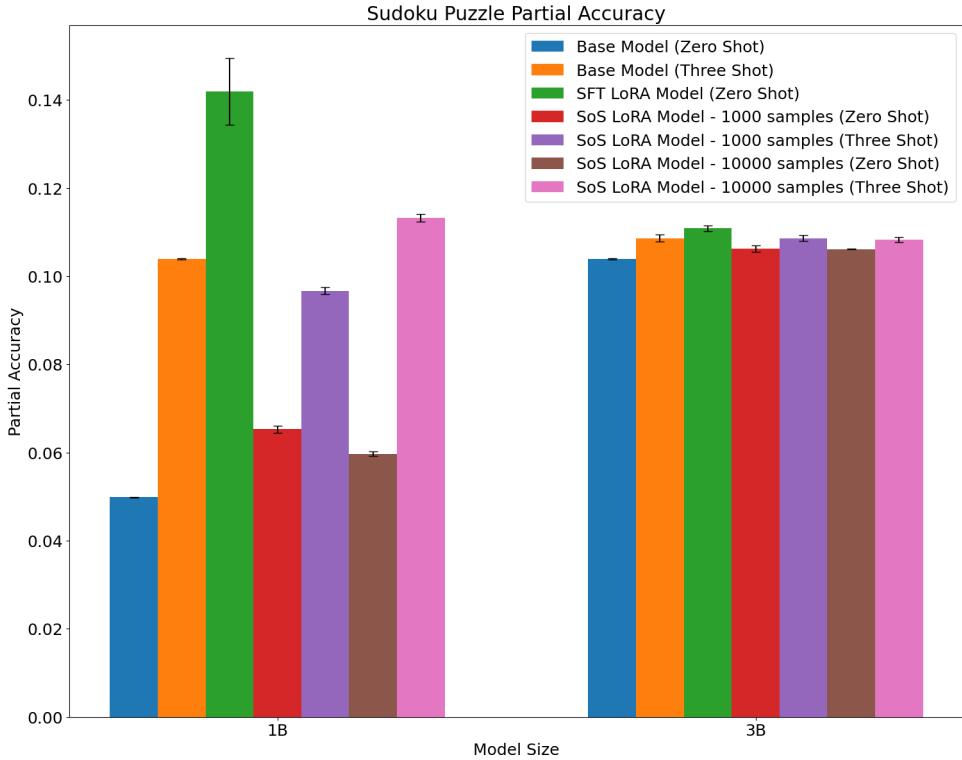


Figure 1: Sudoku Partial Accuracies

269 perform structured reasoning, arithmetic problem-
270 solving, and logical deduction under varied con-
271 straints.

272 Additionally, we train a model using **Count-**
273 **down**, a synthetic dataset of search traces. This
274 dataset was used in Gandhi et al. (2024) to improve
275 performance on search tasks.

276 4.1.1 Sudoku Dataset

277 The Sudoku dataset focuses on structured logical
278 reasoning within a grid-based numerical frame-
279 work. Each sample in the dataset contains:

- 280 • **Puzzle:** A partially filled 9x9 Sudoku grid,
281 where the task is to complete the grid while
282 ensuring that each row, column, and 3x3 sub-
283 grid contains the numbers 1 through 9 without
284 repetition.
- 285 • **Clues:** The number of initially provided filled
286 cells, which determines the problem's com-
287 plexity.
- 288 • **Difficulty:** A numeric rating quantifying the
289 puzzle's challenge based on the number of
290 clues and required logical steps for the solu-
291 tion.
- 292 • **Solution:** The fully solved Sudoku grid.

The dataset³ contains 3,000,000 examples, fil-
tered to ensure high-quality reasoning challenges.
For training and evaluation, the first 10,000 sam-
ples were used, selected for their balanced difficulty
levels and diversity of logical patterns.

4.1.2 Zebra Dataset

The Zebra dataset, also known as the "Zebra Puz-
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1420
1421
1422
1423
1424
1425
1

314	4.1.3 Countdown Dataset	357
315	The Countdown dataset evaluates arithmetic reasoning and problem-solving under constraints. Each sample is a numerical reasoning problem inspired by the classic Countdown game show, where participants manipulate a set of integers using basic arithmetic operations to achieve a target number.	358
316		359
317		360
318		361
319		362
320		363
321	Each instance includes:	
322	<ul style="list-style-type: none"> • Numbers: A set of initial integers available for manipulation. 	364
323		365
324	<ul style="list-style-type: none"> • Target: The number to be achieved through a sequence of operations. 	366
325		367
326	<ul style="list-style-type: none"> • Solution: The optimal sequence of arithmetic operations leading to the target. 	368
327		369
328	<ul style="list-style-type: none"> • Search Path: The trajectory explored during solution generation, including intermediate operations and states. 	370
329		371
330		
331	<ul style="list-style-type: none"> • Rating: A score quantifying the quality of the solution path, derived from a heuristic measure of optimality. 	
332		
333		
334	Using code ⁵ from Gandhi et al. (2024) , we generated a dataset of 500,000 samples. This dataset was then filtered to retain only trajectories with a rating above 0.995. From this curated subset, we trained two models, the first using 1,000 samples and the second using 10,000 samples.	372
335		373
336		374
337		375
338		376
339		377
340	4.2 Dataset Verbalization	378
341	As this work uses instruction-tuned base models, we formatted the datasets into an assistant-user dialog format for supervised fine-tuning. In each case an instance from the dataset would be transformed into the user describing the puzzle and the assistant replying with the solution. Further details on how each dataset was verbalized can be found in Appendix A.	379
342		
343		
344		
345		
346		
347		
348		
349	4.3 Evaluation Metrics	380
350	To assess the performance of our methods, we utilized two evaluation metrics: Partial Accuracy and Strict Accuracy . These metrics were applied to the Sudoku and Zebra datasets, enabling comprehensive evaluation of the model’s reasoning and problem-solving capabilities across varying levels of granularity.	381
351		382
352		383
353		
354		
355		
356		
357	4.3.1 Partial Accuracy	384
358	Partial Accuracy measures the proportion of the solution that is correct, even if the overall solution is incomplete or partially incorrect. This metric is valuable for difficult tasks where models rarely fully solve the puzzle as it provides signal on incremental improvements. For each dataset:	385
359		386
360		387
361		388
362		389
363		390
364	<ul style="list-style-type: none"> • Sudoku: Partial Accuracy evaluates the proportion of correctly filled cells in the 9x9 grid, reflecting the model’s understanding of the underlying logical constraints. 	391
365		392
366		393
367		
368	<ul style="list-style-type: none"> • Zebra: Partial Accuracy measures the percentage of correctly assigned attributes (e.g., house colors, occupants) relative to the total number of attributes to be assigned. 	394
369		395
370		396
371		
372	4.3.2 Strict Accuracy	397
373	Strict Accuracy evaluates the correctness of the solution in its entirety. For a sample to achieve strict accuracy, every component of the solution must be correct without any errors or omissions. This metric is critical for assessing the model’s ability to provide fully valid and coherent outputs.	398
374		399
375		400
376		401
377		402
378		403
379	For each dataset:	404
380		
381	<ul style="list-style-type: none"> • Sudoku: Strict Accuracy requires the entire 9x9 grid to be correctly completed, with all rows, columns, and subgrids satisfying the Sudoku constraints. 	405
382		406
383		407
384	<ul style="list-style-type: none"> • Zebra: Strict Accuracy demands that the complete mapping of entities to attributes satisfies all given logical constraints. 	408
385		409
386		410
387	4.3.3 Rationale for Metric Selection	411
388	These metrics were chosen to capture both incremental and holistic reasoning capabilities:	412
389		413
390	<ul style="list-style-type: none"> • Partial Accuracy: Highlights the model’s progress toward solving a problem, even in cases where it fails to achieve the final correct solution. 	414
391		415
392		416
393		417
394	<ul style="list-style-type: none"> • Strict Accuracy: Ensures that the model is rigorously evaluated on its ability to generate fully correct and valid outputs. 	418
395		419
396		420
397	5 Experimental Details	421
398	Base model weights were downloaded from Huggingface and the LoRA adapters were implemented	422
399		423

allenai/ZebraLogicBench

⁵<https://github.com/kanishkg/stream-of-search>

400 with Huggingface’s Transformers library⁶. The
401 LoRA adapters targeted modules in the self attention
402 mechanisms of the model. All LoRA adapters
403 were trained with the same configuration: with a
404 rank of 16, an α of 32, a dropout rate of 0.05, and
405 no bias.

406 Adapters were trained with Supervised Fine-
407 tuning, with loss calculated only on the model’s
408 completions. We used the SFTTrainer from
409 Huggingface’s TRL library⁷ which uses the
410 AdamW optimizer with an initial learning rate of
411 $2e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e - 8$.

412 To improve result reliability, all experiments
413 were run three times. All runs used a random 80-
414 20 train test split so in each experiment run the
415 adapters were trained and tested on different sub-
416 sets of the datasets.

417 6 Results and Discussion

418 6.1 Results

419 Our experiments evaluated the impact of fine-
420 tuning language models on programmatically gen-
421 erated search traces from the Countdown task, fo-
422 cusing on their ability to generalize to two reason-
423 ing tasks, Sudoku and Zebra puzzles. The evalua-
424 tion metrics used were Partial Accuracy and Strict
425 Accuracy, as described in Section 4 providing com-
426 plementary insights into intermediate reasoning
427 steps and fully correct solutions, respectively.

428 Figures 1, 2, and 3 illustrate the performance
429 across different model configurations and training
430 datasets. Appendix B contains full results in a table
431 for more detailed analysis.

432 6.1.1 Sudoku Partial Accuracy

433 As seen in Figure 1, none of the models tested mod-
434 els performed well on the Sudoku puzzles. For the
435 3B models, Partial Accuracies were at or below
436 11% which is the score you would expect if guess-
437 ing digits randomly. Surprisingly, the 1B parameter
438 fine-tuned model achieved the highest partial accu-
439 racy at around 14%.

440 6.1.2 Sudoku Strict Accuracy

441 Strict Accuracy for all configurations in the Sudoku
442 task was consistently zero. This result shows that
443 this puzzle was too difficult for the models tested,
444 even after fine-tuning.

⁶<https://huggingface.co/docs/transformers/en/index>
⁷<https://huggingface.co/docs/trl/en/index>

445 6.1.3 Zebra Puzzle Partial Accuracy

446 Figure 2 shows that the Zebra fine-tuned model
447 achieved the highest Partial Accuracy, as expected.
448 We find that the base model and SoS-trained mod-
449 els achieve similar performance in most of the trials.
450 We do observe that for the 3B parameter models,
451 the SoS-trained models perform better than the base
452 model in zero shot testing.

453 6.1.4 Zebra Puzzle Strict Accuracy

454 We see that the base and SoS fine-tuned models
455 struggled to achieve non-zero Strict Accuracy in
456 the Zebra task, except in a few cases with the 3B
457 model in three-shot settings. The zebra fine-tuned
458 model (SFT LoRA) achieved the best Strict Accu-
459 racy scores, indicating its superior ability to gener-
460 ate fully correct solutions. Overall Figure 3 shows
461 that all of the models struggled to produce error-
462 free solutions for Zebra puzzles.

463 6.2 Discussion

464 The results highlight critical insights into the
465 strengths and limitations of fine-tuning approaches.

466 6.2.1 Base Fine-Tuned Model Superiority

467 Unsurprisingly, across both tasks and metrics, the
468 base fine-tuned model (SFT LoRA) consistently
469 outperformed the SoS fine-tuned models and base
470 models. This finding confirms that directly fine-
471 tuning on task data is very effective and highlights
472 the challenges of eliciting transfer from synthetic
473 data.

474 6.2.2 Limited Transferability of SoS 475 Fine-Tuning

476 Fine-tuning on SoS data provided some improve-
477 ment in Zebra puzzles, particularly in a zero-shot
478 context. These results indicate that the search sub-
479 skill fine-tuned via SoS data did not generalize well
480 to the broader reasoning demands of Zebra and Su-
481 doku tasks.

482 6.2.3 Impact of Dataset Size, Model Scale and 483 Few-Shot Context

484 Increasing the number of SoS samples from 1,000
485 to 10,000 resulted in incremental gains in Partial
486 Accuracy for Zebra puzzles. Few-shot contexts
487 ($k=3$) further amplified performance improvements
488 for all configurations. Furthermore, larger models
489 (3B) were observed to consistently outperform their
490 smaller counterparts for all configurations.

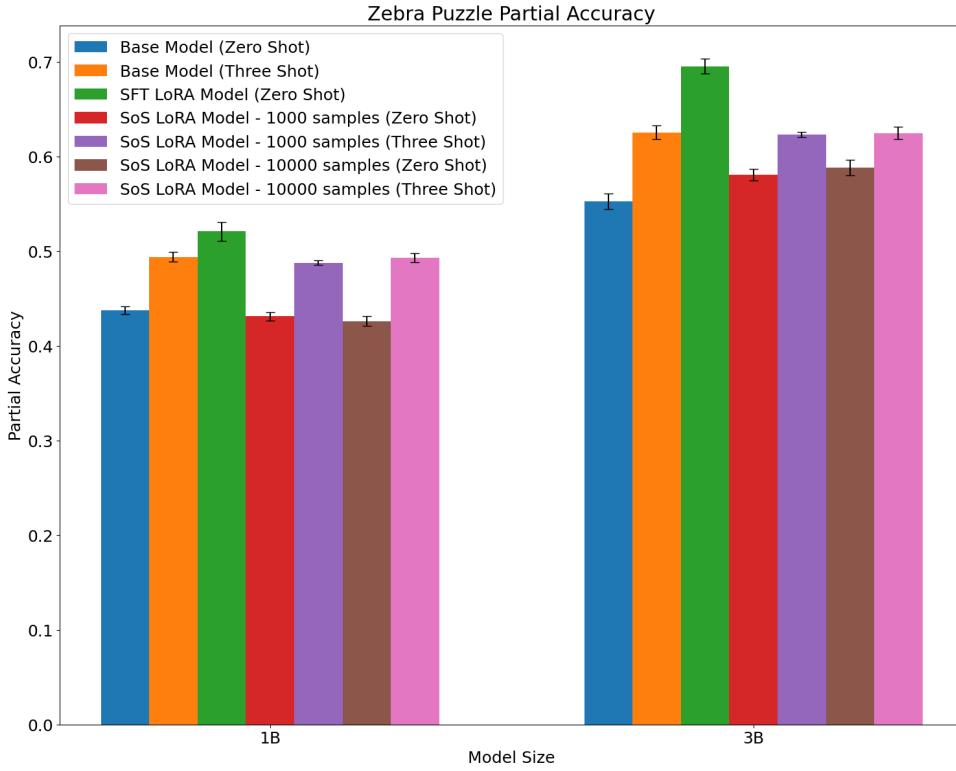


Figure 2: Zebra Partial Accuracies

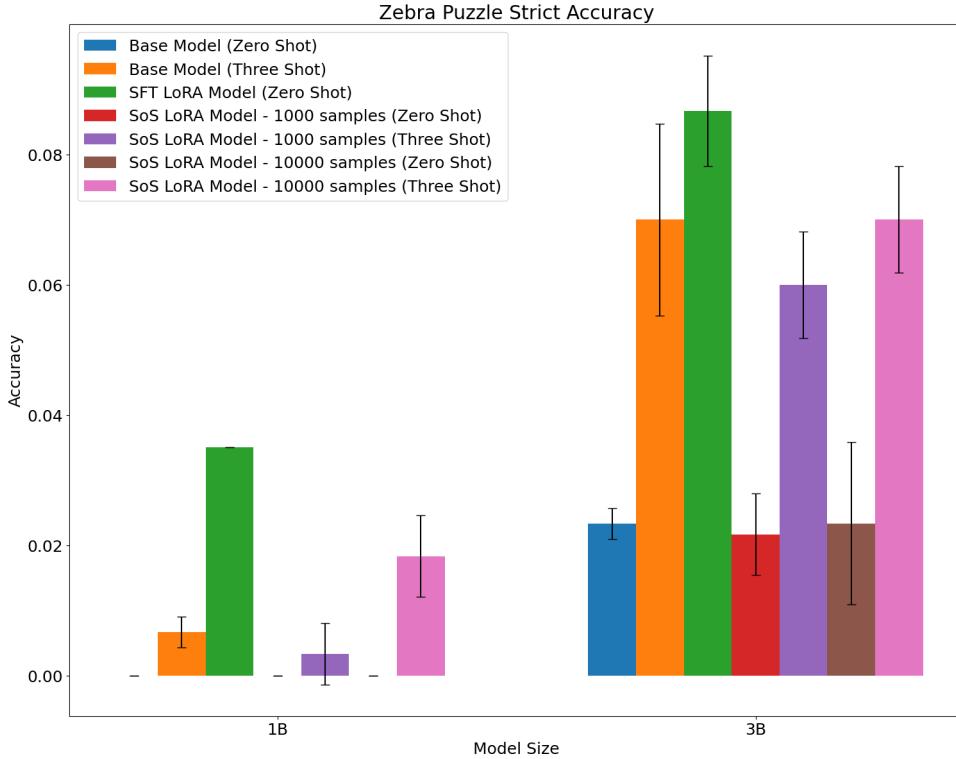


Figure 3: Zebra Strict Accuracies

491 6.2.4 Task-Specific Challenges

492 The stark contrast between Sudoku and Zebra re-
493 sults highlights task-specific demands. Zebra puz-

zles, which rely on natural language reasoning and
494 constraint satisfaction, align more closely with pre-
495 training corpora and thus benefit more from fine-
496

497 tuning. Poor performance across all models shows
498 that Sudoku was too difficult of a downstream task
499 potentially due to the reliance on structured numerical logic.
500

501 7 Conclusion

502 This work explored the effectiveness of fine-tuning
503 large language models on programmatically generated
504 synthetic data, specifically search trajectories, to enhance reasoning sub-skills and evaluate
505 their impact on downstream tasks such as Sudoku
506 and Zebra puzzles. Our findings demonstrate that
507 while synthetic data fine-tuning provides incremental
508 improvements in Partial Accuracy, especially for
509 Zebra puzzles, its ability to transfer to high-level
510 reasoning tasks is limited. Models fine-tuned
511 directly on task-specific datasets consistently outperformed
512 search-fine-tuned models across all metrics and tasks, emphasizing the difficulty in synthetic
513 data approaches. Sudoku was revealed to be
514 too difficult of a task for the models tested as evidenced by Strict Accuracies of 0%. Larger models
515 (3B parameters) and few-shot contexts improved
516 performance, but the observed gains remained task-specific; Zebra puzzles benefited more possibly due
517 to their alignment with natural language reasoning,
518 whereas Sudoku posed a greater challenge. These
519 results highlight the potential of synthetic data for
520 training foundational reasoning sub-skills but suggest
521 that combining it with targeted task-specific
522 fine-tuning may be beneficial in the future.
523

527 7.1 Future Work

528 In the future, work can be done to explore hybrid
529 approaches, expanded reasoning sub-skills, and scaling model size to enhance generalization
530 and improve reasoning performance across diverse
531 tasks. Furthermore, it would be interesting to repeat
532 a similar set of experiments with more reasoning
533 sub-skills such as symbolic reasoning or arithmetic.
534 Expanding the number of sub-skills would allow
535 for analysis of how different sub-skills transfer to
536 different downstream reasoning tasks. Additionally,
537 we would like to increase the size of the base
538 models tested since it is possible that the 1B and
539 3B parameter models used in this work don't have
540 the capacity to learn generalizable sub-skills but
541 that ability may emerge at larger scales.

8 Contributions

Cormac developed the training and evaluation scripts and documented the repository for release. Abhijeet handled the synthetic data generation and built all of the data loaders. Writing the report was a joint effort.

Acknowledgments

Thank you to Xiaoyin Chen for the initial research direction, continued mentorship throughout the project, and providing access to compute resources. Thank to Arkil Patel for valuable feedback on our project proposal. Thank you to Prof. Siva Reddy for an excellent LLMs course and being available as a resource throughout the term.

References

- Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. 2024. [Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling](#).
- Xiang Deng, Yu Su, Alyssa Lees, You Wu, Cong Yu, and Huan Sun. 2021. [ReasonBERT: Pre-trained to reason with distant supervision](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6112–6127, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. 2024. [Stream of search \(sos\): Learning to search in language](#).
- Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. [V-star: Training verifiers for self-taught reasoners](#).
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *CoRR*, abs/2106.09685.
- Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. 2022. [Merit: Meta-path guided contrastive learning for logical reasoning](#).
- Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qin-qing Zheng, Paul Mcvay, Michael Rabbat, and Yandan Tian. 2024. [Beyond a*: Better planning with transformers via search dynamics bootstrapping](#).
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harry Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let's Verify Step by Step](#). ArXiv:2305.20050.

- 592 Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin,
593 Qiang Fu, Yan Gao, Jian-Guang Lou, and Weizhu
594 Chen. 2022. [Reasoning like program executors](#).
- 595 Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Pan-
596 igrahy. 2024. [Causal language modeling can elicit](#)
597 [search and reasoning capabilities on logic puzzles](#).
- 598 Avi Singh, John D. Co-Reyes, Rishabh Agarwal,
599 Ankesh Anand, Piyush Patil, Xavier Garcia, Pe-
600 ter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu,
601 Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex
602 Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet,
603 Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch,
604 Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jef-
605 frey Pennington, Jiri Hron, Kathleen Kenealy, Kevin
606 Swersky, Kshitij Mahajan, Laura Culp, Lechao
607 Xiao, Maxwell L. Bileschi, Noah Constant, Roman
608 Novak, Rosanne Liu, Tris Warkentin, Yundi Qian,
609 Yamini Bansal, Ethan Dyer, Behnam Neyshabur,
610 Jascha Sohl-Dickstein, and Noah Fiedel. 2024. [Be-](#)
611 [yond human data: Scaling self-training for problem-](#)
612 [solving with language models](#).
- 613 Kaya Stechly, Karthik Valmeekam, and Subbarao Kamb-
614 hampati. 2024. [Chain of thoughtlessness? an analy-](#)
615 [sis of cot in planning](#).
- 616 Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan,
617 and Subbarao Kambhampati. 2022. [Large language](#)
618 [models still can't plan \(a benchmark for LLMs on](#)
619 [planning and reasoning about change\)](#). In *NeurIPS*
620 *2022 Foundation Models for Decision Making Work-*
621 *shop*.
- 622 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc
623 Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery,
624 and Denny Zhou. 2023. [Self-Consistency Improves](#)
625 [Chain of Thought Reasoning in Language Models](#).
626 ArXiv:2203.11171.
- 627 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten
628 Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le,
629 and Denny Zhou. 2023. [Chain-of-Thought Prompt-](#)
630 [ing Elicits Reasoning in Large Language Models](#).
631 ArXiv:2201.11903.
- 632 Michihiro Yasunaga, Jure Leskovec, and Percy Liang.
633 2022. [Linkbert: Pretraining language models with](#)
634 [document links](#).
- 635 Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-
636 Zhu. 2024. [Physics of language models: Part 2.2,](#)
637 [how to learn from mistakes on grade-school math](#)
638 [problems](#).
- 639 Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024.
640 [Distilling system 2 into system 1](#).

641 **A Dataset Verbalization Format**

642 **A.1 Sudoku**

Sudoku Dataset Verbalized Input

[User]: Given the Sudoku puzzle 1..5.37..6.3..8.9.....98...1.....8761.....6.....7.8.9.76.47...6.312, which has 27 clues and a difficulty rating of 2.2. Please solve for the final arrangement.

643

Sudoku Dataset Verbalized Output

[Assistant]: 198543726643278591527619843 914735268876192435235486179
462351987381927654759864312

644

645 **A.2 Zebra**

Zebra Dataset Verbalized Input

[User]: Given there are 2 houses, numbered 1 to 2 from left to right. Each house is occupied by a different person. Each house has a unique attribute for each of the following characteristics:

- Each person has a unique name: Arnold, Eric
- People own unique car models: ford f150, tesla model 3
- The people keep unique animals: cat, horse

Clues:

1. Eric is directly left of the person who owns a Tesla Model 3.
2. The person who keeps horses is in the first house.

Please solve for the final arrangement.

646

Zebra Dataset Verbalized Output

[Assistant]: The solution is as follows:

In house 1, name is Eric, car model is ford f150, and animal is horse.

In house 2, name is Arnold, car model is tesla model 3, and animal is cat.

647

648 **A.3 Countdown**

Countdown Dataset Verbalized Input

[User]: You are tasked to solve arithmetic reasoning problems. Given a set of numbers and a target, describe the steps in the path to reach the target using those numbers. Using arithmetic operations such as addition (+), subtraction (-), multiplication (*) and division (/), use the initial set of numbers to gather new numbers that eventually reach the target in the end. The target is: 48 The available numbers are: [12, 96, 68, 32] Describe how to reach the target using the given numbers.

649

Countdown Dataset Verbalized Output

[Assistant]: The search path used for this problem was: Current State: 48:[12, 96, 68, 32], Operations: [] Exploring Operation: 96-68=28, Resulting Numbers: [12, 32, 28] Generated Node #0,0: 48:[12, 32, 28] Operation: 96-68=28 Moving to Node #0,0 Current State: 48:[12, 32, 28], Operations: ['96-68=28'] Exploring Operation: 32-28=4, Resulting Numbers: [12, 4] Generated Node #0,0,0: 48:[12, 4] Operation: 32-28=4 Moving to Node #0,0,0 Current State: 48:[12, 4], Operations: ['96-68=28', '32-28=4'] Exploring Operation: 12+4=16, Resulting Numbers: [16] 16,48 unequal: No Solution Moving to Node #0,0,0 Current State: 48:[12, 4], Operations: ['96-68=28', '32-28=4'] Exploring Operation: 12*4=48, Resulting Numbers: [48] 48,48 equal: Goal Reached

The optimal path was: Current State: 48:[12, 96, 68, 32], Operations: [] Exploring Operation: 96-12=84, Resulting Numbers: [68, 32, 84] Generated Node #2: [68, 32, 84] from Operation: 96-12=84 Current State: 48:[68, 32, 84], Operations: ['96-12=84'] Exploring Operation: 68-32=36, Resulting Numbers: [84, 36] Generated Node #3: [84, 36] from Operation: 68-32=36 Current State: 48:[84, 36], Operations: ['96-12=84', '68-32=36'] Exploring Operation: 84-36=48, Resulting Numbers: [48] 48,48 equal: Goal Reached

The final solution was: 96-12=84, 68-32=36, 84-36=48

650

B Results Tables

651

B.1 Full Sudoku Results

652

Run Configuration	Partial Accuracy	Strict Accuracy
Base Evaluation Runs		
Zero-Shot, 1B Model	0.0499 (0.00004)	0.0 (0.0)
Three-Shot, 1B Model	0.104 (0.0001)	0.0 (0.0)
Zero-Shot, 3B Model	0.104 (0.0001)	0.0 (0.0)
Three-Shot, 3B Model	0.109 (0.0008)	0.0 (0.0)
Fine-Tuned Training Runs		
Zero-Shot, 1B Model	0.111 (0.008)	0.0 (0.0)
Zero-Shot, 3B Model	0.142 (0.0006)	0.0 (0.0)
SOS Evaluation - 1000 Samples		
Zero-Shot, 1B Model	0.065 (0.0008)	0.0 (0.0)
Three-Shot, 1B Model	0.097 (0.0008)	0.0 (0.0)
Zero-Shot, 3B Model	0.106 (0.0007)	0.0 (0.0)
Three-Shot, 3B Model	0.109 (0.0007)	0.0 (0.0)
SOS Evaluation - 10000 Samples		
Zero-Shot, 1B Model	0.059 (0.0005)	0.0 (0.0)
Three-Shot, 1B Model	0.113 (0.0008)	0.0 (0.0)
Zero-Shot, 3B Model	0.106 (0.00008)	0.0 (0.0)
Three-Shot, 3B Model	0.108 (0.0006)	0.0 (0.0)

653 **B.2 Full Zebra Puzzle Results**

Run Configuration	Partial Accuracy	Strict Accuracy
Base Evaluation Runs		
Zero-Shot, 1B Model	0.438 (0.004)	0.0 (0.0)
Three-Shot, 1B Model	0.4947 (0.005)	0.007 (0.002)
Zero-Shot, 3B Model	0.553 (0.008)	0.023 (0.002)
Three-Shot, 3B Model	0.625 (0.007)	0.07 (0.014)
Fine-Tuned Training Runs		
Zero-Shot, 1B Model	0.521 (0.010)	0.035 (0.0)
Zero-Shot, 3B Model	0.695 (0.008)	0.087 (0.008)
SOS Evaluation - 1000 Samples		
Zero-Shot, 1B Model	0.431 (0.005)	0.0 (0.0)
Three-Shot, 1B Model	0.488 (0.002)	0.003 (0.005)
Zero-Shot, 3B Model	0.581 (0.006)	0.022 (0.006)
Three-Shot, 3B Model	0.623 (0.003)	0.06 (0.008)
SOS Evaluation - 10000 Samples		
Zero-Shot, 1B Model	0.426 (0.005)	0.0 (0.0)
Three-Shot, 1B Model	0.493 (0.005)	0.018 (0.006)
Zero-Shot, 3B Model	0.588 (0.008)	0.023 (0.012)
Three-Shot, 3B Model	0.625 (0.007)	0.07 (0.008)