

000 001 002 003 004 005 006 007 CRITIQUE-RL: TRAINING CRITIQUING LANGUAGE 008 MODELS THROUGH TWO-STAGE RL FOR IMPROVED 009 DISCRIMINATION AND CONSTRUCTIVE FEEDBACK 010 011

012 **Anonymous authors**
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
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
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
19

054	Problem: If a bag of marbles costs \$20 and the price increases by 20% of the original price every two months, how much would a bag of marbles cost after 36 months?
055	Actor Model (original response): Let's break it down step by step: 1. The price of the bag of marbles increases by $20/100 * \$20 = \4 every two months. 2. After 36 months, the price of the bag of marbles will have increased $36/(2*2) = 9$ times. 3. The price of the bag of marbles will have increased by $\$4, 9$ times for a total of $\$4 * 9 = \36 . 4. The cost of a bag of marbles after 36 months will be $\$20 + \$36 = \$56$. The answer is 56. Correctness Verifier: <input checked="" type="checkbox"/>
056	Critique Model (critique): Step sentence: The price of the bag of marbles increases by $20/100 * \$20 = \4 every two months. Correctness of the step: Correct Explanation: This step correctly calculates the increase in price every two months, which is 20% of the original price of \$20. The calculation is $0.20 * \$20 = \4 .
057	Step sentence: After 36 months, the price of the bag of marbles will have increased $36/(2*2) = 9$ times. Correctness of the step: Wrong Explanation: This step is wrong. The price increases every two months, so the number of increases in 36 months is $36 / 2 = 18$, not $36 / (2 * 2) = 9$. There are 18 two-month periods in 36 months. ... Correctness of the final answer: Wrong Discrimination Verifier: <input checked="" type="checkbox"/>
058	Actor Model (refinement response): Let's break it down step by step: To determine the cost of a bag of marbles after 36 months, we need to consider the following: 1. Initial Price: The initial price of the bag of marbles is \$20. 2. Price Increase Calculation: The price increases by 20% of the original price every two months. The increase in price every two months is calculated as $0.20 * \$20 = \4 . 3. Number of Increases: The price increases every two months, and there are 36 months in total. The number of increases in 36 months is $36 / 2 = 18$. 4. Total Increase in Price: The total increase in price over 36 months is calculated by multiplying the number of increases by the increase per period. Total increase = $\$4 * 18 = \72 . 5. Final Price Calculation: The final price of the bag of marbles after 36 months is the initial price plus the total increase. Final price = $\$20 + \$72 = \$92$. The answer is 92. Correctness Verifier: <input checked="" type="checkbox"/>
059	
060	
061	
062	
063	
064	
065	
066	
067	
068	
069	
070	
071	
072	
073	
074	
075	

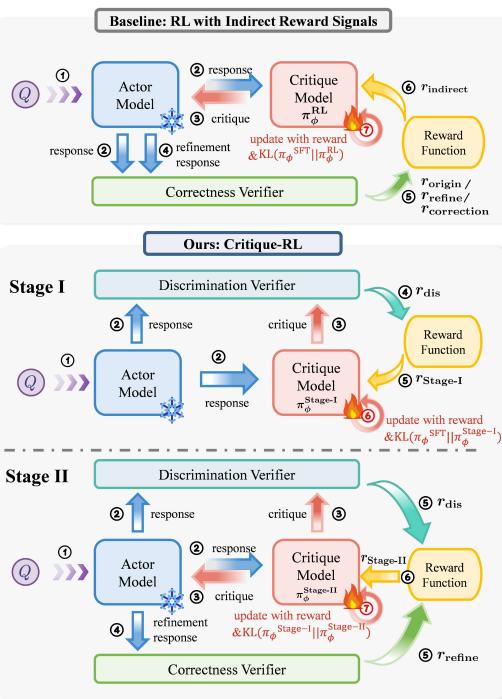


Figure 2: **Left:** A case illustrating the two-player actor-critic interaction, including the original response from the actor, the critique from the critic, and the refinement from the Actor. **Right:** Overview of our method and its comparison with baseline RL. The snowflake icon  on the Actor indicates that it is fixed, while the fire icon  on the Critic indicates that it will be updated. Our method employs a two-stage RL process. It optimize discriminability of critique models in Stage I, and optimize helpfulness while maintaining discriminability in Stage II.

the learner’s output distribution (Kumar et al., 2024). Another line of work does not train the model but instead relies on the model’s inherent abilities, using prompt engineering to elicit its critiquing abilities (Bai et al., 2022; Madaan et al., 2023; Dhuliawala et al., 2024). However, such methods typically assume an oracle verifier during testing, allowing the critique model to bypass discrimination (i.e., determining whether a response is high-quality) and instead focus only on offering helpful feedback for revision (Xi et al., 2024; Gou et al., 2024). Without the oracle verifier, they often meet performance bottleneck (Huang et al., 2024).

In this work, we aim to develop critiquing language models without relying on stronger labeling or an oracle reward function during testing. To this end, we propose Critique-RL, an online RL approach based on two-player actor-critic interaction (Yao et al., 2024; Xi et al., 2024) for developing critique models. In our approach, there are two main roles: the actor and critic. The critic assesses (discriminability) and provides natural language feedback (helpfulness) for the actor’s output, and the actor performs refinement accordingly (Saunders et al., 2022).

To build our method, we first use the correctness of the actor’s two attempts to shape the reward signals for the RL optimization of critique models (§4.1), following approaches like Retroformer (Yao et al., 2024) and CTRL (Xie et al., 2025), where such indirect signals are shown to reflect the quality of critiques. However, this approach fails to develop satisfactory critique models, i.e., with low performance. Delving into the optimization process, we reveal that while the helpfulness of the critique models improves, their discriminability is not well optimized, leading to an optimization bottleneck and even a collapse of RL training.

To address the challenges, Critique-RL employs a two-stage RL approach (§4.2). Specifically, as shown in Figure 2, in the first stage, we optimize the discriminability of the critique models using direct rule-based reward signals. In the second stage, we introduce indirect rewards based on the correctness of actor refinement to enhance the helpfulness, while using appropriate regularization to maintain their discriminability. In-depth training dynamics shows that our method addresses the

108 training collapse and stably optimizes both discriminability and helpfulness. Extensive experiments
 109 show that our method outperforms baselines across different models and tasks, yielding a 9.02%
 110 improvement on in-domain tasks and 5.70% improvement on out-of-domain tasks for Qwen2.5-7B.
 111 It is also noteworthy that critique models trained with our method can generalize to unseen tasks,
 112 demonstrating its promise for scalable oversight.

113 In summary, our main contributions are:
 114

- 115 1. Delving into the RL optimization process, we reveal that solely depending on indirect reward
 116 signals of actor’s output correctness cannot develop effective critique models, which poses conflict
 117 and optimization challenges between the discriminative and feedback capabilities of critics.
- 118 2. We then propose Critique-RL, a novel two-stage RL approach to develop critique models for
 119 providing accurate assessment and helpful feedback for model outputs.
- 120 3. We perform in-depth experiments, ablation and analysis to show the effectiveness and stability of
 121 our method. We hope our work provides insights for the community.

124 2 RELATED WORK

126 **Prompt engineering for eliciting critiquing ability from language models.** As a key technique
 127 for scalable oversight (Bowman et al., 2022), many previous works have explored the use of prompt
 128 engineering to elicit the critiquing and reflection abilities of LLMs (Bai et al., 2022; Madaan et al.,
 129 2023; Ye et al., 2023; Dhuiliawala et al., 2024). These methods typically rely on an oracle verifier
 130 including answer matching or external tools at test time for discrimination, allowing the LLM to
 131 focus solely on providing natural language feedback (Xi et al., 2024; Huang et al., 2024). However,
 132 in the absence of an external verifier, even SOTA models face significant challenges (Saunders et al.,
 133 2022; Welleck et al., 2023; Xu et al., 2024; Huang et al., 2024). In this work, we do not assume an
 134 oracle verifier; instead, we train critique models through RL to optimize both discriminability and the
 135 ability to provide helpful feedback.

136 **Fine-tuning language models for critiquing.** Previously, a line of work has explored fine-tuning-
 137 based approaches for training critique models (Saunders et al., 2022; Bowman et al., 2022; Xi et al.,
 138 2024). However, these methods primarily rely on a [stronger supervisor for data annotation](#), which is
 139 costly and difficult to scale (Xi et al., 2024). To address this issue, some researchers have proposed
 140 self-improvement-based methods to train models for self-critiquing (Tang et al., 2025; Zheng et al.,
 141 2024; Yuan et al., 2025). Unlike these approaches, we adopt a two-player paradigm and train a
 142 separated critique model through RL.

144 **Reinforcement learning for language models.** RL has become an essential component of LLM
 145 post-training, such as RLHF for alignment (Ouyang et al., 2022; Zheng et al., 2023; Wang et al.,
 146 2024; Shao et al., 2024). Additionally, various works have leveraged RL to enhance language models’
 147 performance in reasoning (Snell et al., 2024; Kumar et al., 2024), coding (Kumar et al., 2024), and
 148 decision-making tasks (Shinn et al., 2023). Furthermore, some studies explore using RL to improve
 149 LM’s ability for self-reflection and self-correction (McAleese et al., 2024; Kumar et al., 2024; Welleck
 150 et al., 2023; Shinn et al., 2023; Xu et al., 2024; Ye et al., 2023). Other methods, such as Retroformer
 151 (Yao et al., 2024) and CTRL (Xie et al., 2025), leverage indirect reward signals to optimize critique
 152 model’s helpfulness, targeting decision-making tasks and coding tasks, respectively. However, their
 153 RL phase overlooks the joint optimization of discriminability and helpfulness. Different from them,
 154 we propose a two-stage Critique-RL approach to optimize both discriminability and helpfulness,
 155 effectively developing critique models.

157 3 PRELIMINARIES

159 3.1 THE TWO-PLAYER INTERACTION FRAMEWORK

161 The multi-agent framework in this work consists of two main roles (Yao et al., 2024; Xi et al., 2024):
 162 the actor model and the critique model. It operates through a response-critique-refinement process.

162 Specifically, given a question x , the actor model is expected to generate an original response $y = \pi_\theta(x)$, which includes both the reasoning trajectory and the final answer. The correctness verifier then
 163 provides an oracle reward $r_{\text{oracle}}(x, y)$ to the actor model. Subsequently, the critique model π_ϕ takes
 164 the question-response pair (x, y) as input and produces critique $c = \pi_\phi(x, y)$, which should include
 165 assessment of the response correctness (discriminability) and offer constructive natural language
 166 feedback (helpfulness). Based on this critique, the actor model generates a refinement response
 167 $y' = \pi_\theta(x, y, c)$, and subsequently receives an oracle reward $r_{\text{oracle}}(x, y')$. Using these rewards,
 168 i.e., $r_{\text{oracle}}(x, y)$ and $r_{\text{oracle}}(x, y')$, we can design different reward functions $r_c(\cdot)$ for critique models,
 169 which will be shown in §4.
 170

172 3.2 POLICY GRADIENT FOR LLMs

173 Policy gradient methods (Sutton et al., 1999), e.g., REINFORCE (Ahmadian et al., 2024; Kumar
 174 et al., 2024), are common techniques to perform RL on LLMs (Ouyang et al., 2022). For the policy
 175 critique model π_ϕ parameterized by ϕ , the objective of policy gradient is to find an optimal policy
 176 that maximizes the reward function $r_c(\cdot)$. It is typically expressed as maximizing:
 177

$$178 \mathbb{E}_{c \sim \pi_\phi(\cdot|x, y), y' \sim \pi_\theta(x, y, c)} [r_c(x, y, c, y')], \quad (1)$$

179 where $\mathbb{E}_{c \sim \pi_\phi(\cdot|x, y), y' \sim \pi_\theta(x, y, c)}$ denotes the expectation over the critique sampled from the critic π_ϕ
 180 and the refinement response sampled from the actor π_θ . This gradient is used to optimize the critique
 181 model via gradient ascent. The positive critique is “reinforced” by increasing its probability.
 182

183 3.3 EVALUATION METRICS

184 To evaluate the performance of the critique model, we consider the following metrics: (1)
 185 **Acc@Refine**: the accuracy of the actor model’s refinement response; (2) Δ : the improvement
 186 in the actor model’s accuracy between the original and refinement response, which measures the
 187 effectiveness of the critique model; (3) $\Delta^{c \rightarrow i}$: the change rate from an originally correct response
 188 to an incorrect refinement response. A lower value is better; (4) $\Delta^{i \rightarrow c}$: the change rate from an
 189 originally incorrect response to a correct refinement response. A higher value is better; (5) **Acc@Dis**:
 190 a direct metric to measure the discriminability of the critique model, which quantifies the accuracy of
 191 whether the correctness accessed by the critic aligns with the true correctness of the original response.
 192

193 4 METHODOLOGY

194 4.1 MOTIVATING FINDINGS: RL WITH INDIRECT REWARD SIGNALS IS INSUFFICIENT FOR 195 TRAINING SATISFACTORY CRITIQUE MODELS

196 In the two-player actor-critic framework (Yao et al., 2024; Xi et al., 2024), a natural and intuitive way
 197 to optimize the critiquing language models is to shape the reward signals derived from the actor’s
 198 two attempts (original and refinement responses). We explore several reward shaping approaches,
 199 demonstrate their failure modes, and investigate why they fail to incentivize satisfactory critiquing
 200 ability.
 201

202 **Analysis setups: data, models, and training methods.** Our preliminary experiments are on
 203 GSM8K (Cobbe et al., 2021), and the backbone model is Qwen2.5-3B (Team, 2024). Following
 204 previous work (Xi et al., 2024), we train an actor model capable of generating responses and
 205 **reasonably following critiques**. To build the SFT dataset for initializing a base critique model,
 206 we prompt Qwen2.5-3B-Instruct to obtain critique data $\mathcal{D}_{\text{SFT}} = \{x, y, c\}_{i=1}^{|\mathcal{D}_{\text{SFT}}|}$, rather than using
 207 annotations from SOTA commercial models like GPT-4o (OpenAI, 2023). We filter the critique data
 208 based on the correctness of refinement to ensure the quality.
 209

210 Next, we train the critique model π_ϕ using the SFT loss:

$$211 \mathcal{L}_{\text{SFT}}(\phi) = \mathbb{E}_{(x, y, c) \sim \mathcal{D}_{\text{SFT}}} \left[\log \pi_\phi(c|x, y) \right]. \quad (2)$$

212 We then employ policy gradient (Sutton et al., 1999) to maximize:

$$213 \mathbb{E}_{c \sim \pi_\phi^{\text{RL}}(\cdot|x, y), y' \sim \pi_\theta(\cdot|x, y, c)} \left[r_c(x, y, c, y') - \beta \text{KL}(\pi_\phi^{\text{SFT}}(c|x, y) || \pi_\phi^{\text{RL}}(c|x, y)) \right], \quad (3)$$

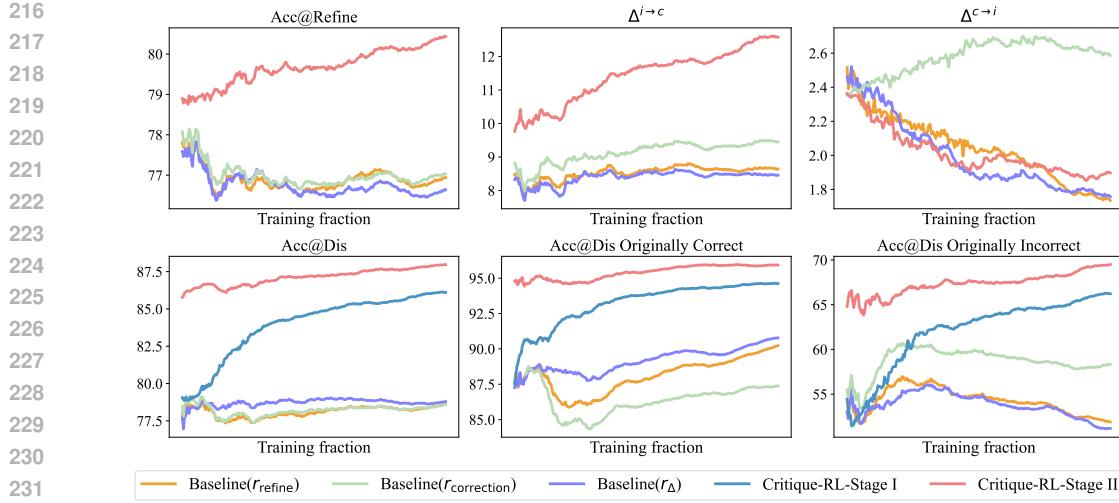


Figure 3: Training dynamics of preliminary experiments. “Acc@Dis Originally Correct” and “Acc@Dis Originally Incorrect” refer to the discrimination accuracy of originally correct and incorrect responses, respectively. Baselines using indirect reward signals to optimize helpfulness tend to exhibit overly conservative or aggressive behavior as the discriminability is not well optimized. In contrast, our Critique-RL optimizes discriminability in Stage I, and optimizes helpfulness while maintaining discriminability in Stage II, achieving better in Acc@Refine, $\Delta^{c \rightarrow i}$ and $\Delta^{i \rightarrow c}$.

where π_θ is the fixed actor model, π_ϕ^{SFT} is the SFT model. Each x is a query sampled from the RL dataset \mathcal{D}_{RL} , y is the original response. $\text{KL}(\cdot || \cdot)$ means the KL-divergence which constrains the distance between the RL model and the SFT model, and β is a scaling factor. $r_c(\cdot)$ is the reward function for critique models. Here, with r_{oracle} being the oracle reward function that verifies the correctness of an actor response, $r_c(\cdot)$ can be r_{refine} which represents the correctness of the refinement:

$$r_{\text{refine}}(x, y, c, y') = r_{\text{oracle}}(x, y'), \quad (4)$$

or it can be r_Δ which represents the difference in correctness between the actor’s two attempts:

$$r_\Delta(x, y, c, y') = r_{\text{oracle}}(x, y') - r_{\text{oracle}}(x, y). \quad (5)$$

Moreover, we also include $r_{\text{correction}}$ as $r_c(\cdot)$ for reinforcing the ability to correct incorrect responses:

$$r_{\text{correction}}(x, y, c, y') = \begin{cases} 1.0, & r_{\text{oracle}}(x, y) = 0 \text{ and } r_{\text{oracle}}(x, y') = 1, \\ 0.2, & r_{\text{oracle}}(x, y) = 1 \text{ and } r_{\text{oracle}}(x, y') = 1, \\ 0.0, & r_{\text{oracle}}(x, y') = 0. \end{cases} \quad (6)$$

Empirical findings and behavior analysis. We illustrate the training dynamics during RL in Figure 3. Optimizing with r_{refine} and r_Δ can reduce $\Delta^{c \rightarrow i}$, preventing originally correct responses from being altered incorrectly, but its $\Delta^{i \rightarrow c}$ is not significantly optimized, meaning its error correction performance is not good enough. This phenomenon reveals that the critique model is overly **conservative**, encouraging the actor to not change its answers. As a result, the final Acc@Refine is not satisfactory.

In contrast, optimizing with $r_{\text{correction}}$ improves $\Delta^{i \rightarrow c}$, but fails to effectively reduce $\Delta^{c \rightarrow i}$. This means it often provides more **aggressive** suggestions, encouraging the actor model to correct incorrect responses, but it also introduces a greater risk of turning originally correct answers into incorrect ones. Similarly, the final Acc@Refine is also not satisfactory.

Analyzing underlying reasons for the failure modes. To reveal the reasons behind the above failure modes, we also visualize the discrimination performance of the critiquing language models during RL in Figure 3. We find that as RL progresses, all three reward functions r_{refine} , r_Δ and $r_{\text{correction}}$ fail to optimize discriminability effectively. For originally correct and incorrect responses, they can only optimize the judgment for one, while the ability to judge the other is reduced. This may be because both of the indirect reward functions are based on the actor’s responses, targeting helpfulness and overlooking discriminability. This motivates the proposal of our method.

270 4.2 TWO-STAGE CRITIQUE-RL
271

272 **Key challenges.** Based on the previous analysis, we have identified two key challenges in RL for
273 critiquing language models: (1) optimizing the discriminability of critique models to improve their
274 accuracy in judging both correct and incorrect original responses; (2) improving the quality of the
275 model’s feedback, i.e., helpfulness, while maintaining its discriminability, to prevent the issues of
276 being overly aggressive or overly conservative.

277 **Method overview.** To address the above challenges, we propose the two-stage Critique-RL. In the
278 first stage, our method explicitly optimizes the discriminability of the critique model using direct
279 reward signals. We then use the resulting model $\pi_{\phi}^{\text{Stage-I}}$ as the initialization for the second stage. In
280 the second stage, we introduce a reward function based on the actor’s response to optimize the critic’s
281 helpfulness, while also incorporating appropriate regularization to maintain its discriminability. We
282 illustrate our method in Figure 2 and the algorithm is summarized in Algorithm 1.
283

284 **Stage I: optimizing discriminability through direct reward signals.** We decouple the discriminability
285 and helpfulness of the critique models (Saunders et al., 2022). In Stage I, we shape the
286 reward based solely on the actor’s original response. Given (x, y) , critique models are prompted to
287 give correctness judgments for each step, and also provide a judgment for the final answer. Based on
288 this, we define the discriminability reward function of the critique models as:
289

$$r_{\text{dis}}(x, y, c) = \mathbb{1}(f(x, y, c) = r_{\text{oracle}}(x, y)), \quad (7)$$

290 where $f(x, y, c)$ is the critique model’s judgment of the correctness of the original response. $\mathbb{1}(\cdot)$ is
291 indicator function that returns 1 only when the condition inside the parentheses holds, and 0 otherwise.
292 Based on this, our Stage I RL maximizes:
293

$$\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-I}}(\cdot|x, y)} \left[r_{\text{dis}}(x, y, c) - \beta \text{KL}(\pi_{\phi}^{\text{SFT}}(c|x, y) || \pi_{\phi}^{\text{Stage-I}}(c|x, y)) \right], \quad (8)$$

294 where the KL divergence with the SFT model is still used to stabilize the training. As shown in Figure
295 3, our Stage I RL can effectively and stably optimize discriminability, regardless of the correctness of
296 the original response.
297

300 **Stage II: optimizing helpfulness while maintaining discriminability.** The goal of the second
301 stage of Critique-RL is to optimize the helpfulness of the critique models without sacrificing their
302 discriminability, thereby avoiding overly conservative or overly aggressive behavior patterns. To
303 achieve this, we introduce a reward function r_{refine} based on actor refinement correctness. Meanwhile,
304 to preserve the model’s discriminability, we retain r_{dis} and introduce a regularization term based on
305 the KL divergence with the Stage I model $\pi_{\phi}^{\text{Stage-I}}$. Specifically, we maximize the following objective:
306

$$\mathbb{E}_{c \sim \pi_{\phi}^{\text{Stage-II}}(\cdot|x, y), y' \sim \pi_{\theta}(\cdot|x, y, c)} \left[r_{\text{refine}} + \beta_1 r_{\text{dis}}(x, y, c) - \beta_2 \text{KL}(\pi_{\phi}^{\text{Stage-I}}(c|x, y) || \pi_{\phi}^{\text{Stage-II}}(c|x, y)) \right], \quad (9)$$

307 where β_1 and β_2 are scaling factors. As shown in Figure 3, our Stage II effectively optimizes
308 the model’s helpfulness, increasing $\Delta^{i \rightarrow c}$ and decreasing $\Delta^{c \rightarrow i}$, ultimately leading to a stable
309 improvement in Acc@Refine and Δ . Our method also performs strongly on the test set (see §5).
310

312 5 EXPERIMENTS
313314 5.1 EXPERIMENTAL SETUP
315

316 **Datasets.** Focusing on mathematical reasoning tasks, we select 5 different commonly-used tasks,
317 including free-from and multiple-choice. Following Ding et al. (2025), we construct training set with
318 the train-split of MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AQUA (Ling et al.,
319 2017). The testset of the three tasks are used as in-domain testset, while the test-split of SVAMP
320 (Patel et al., 2021), TheoremQA (Chen et al., 2023), are used as our OOD (out-of-domain) testset.
321

322 **Models and baselines.** Our experiments are mainly conducted on Qwen2.5 series (Team, 2024),
323 i.e., Qwen2.5-3B and Qwen2.5-7B. Besides, we also conduct experiments on other models like
324 Qwen2.5-72B, Llama3.2 (Dubey et al., 2024) and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI,
325 2025) (see Appendix D and Section 6). We include several baselines: (1) SFT which fine-tunes

324
 325 Table 1: Main results. The best performance is in **bold** and underlined, while the second-best
 326 performance is underlined. Our method is marked in **blue**. No Critic means the actor model
 327 perform reasoning only, and we report the reasoning performance. For other methods, we report the
 328 Acc@Refine performance for the acc column.

329 Model	330 Method	331 MATH			332 GSM8K			333 AQuA		
		334 Acc	335 Δ	336 Acc@Dis	337 Acc	338 Δ	339 Acc@Dis	340 Acc	341 Δ	342 Acc@Dis
343 Qwen2.5-3B	No Critic	36.90	—	—	66.03	—	—	50.00	—	—
	SFT	44.24	7.34	66.51	69.14	3.11	76.34	46.46	-3.54	61.97
	STaR	44.38	7.48	66.97	71.95	<u>5.91</u>	74.79	50.39	0.39	<u>66.13</u>
	Retroformer	44.54	7.64	65.11	70.51	4.47	<u>77.59</u>	51.18	1.18	58.44
	CTRL	46.14	9.24	69.29	70.58	4.55	76.71	53.54	3.54	62.20
	Critique-RL	48.60	11.70	82.80	75.89	9.86	87.44	56.69	6.69	69.92
344 Qwen2.5-7B	No Critic	45.74	—	—	75.66	—	—	63.39	—	—
	SFT	51.84	6.10	67.59	78.77	3.11	79.42	59.45	-3.94	68.67
	STaR	<u>54.06</u>	<u>8.32</u>	69.71	80.52	4.85	81.03	57.87	-5.51	<u>72.18</u>
	Retroformer	52.34	6.60	68.03	80.82	5.16	77.05	63.39	0.00	70.56
	CTRL	53.86	8.12	<u>71.42</u>	81.35	<u>5.69</u>	83.44	64.96	<u>1.57</u>	71.66
	Critique-RL	58.40	12.66	85.20	87.72	12.05	90.43	65.75	2.36	78.09

345 models with critique data. (2) STaR (Zelikman et al., 2022) which iteratively fine-tunes critique
 346 models on self-generated data and filtered based on the refinement correctness of the actor. (3) RL
 347 baselines that leverages indirect outcome-based reward as baselines, i.e., Retroformer (Yao et al.,
 348 2024) which uses PPO and CTRL (Xie et al., 2025) which uses GRPO.

349 **Implementation details.** All experiments are conducted on 8 NVIDIA A800 GPUs. To initialize
 350 an actor that can reason and refine based on the critiquing feedback, we follow Ding et al. (2025); Xi
 351 et al. (2024) to construct a dataset of 21,973 reasoning traces and 12,000 refinement responses. For
 352 critique data, we construct a set of 6,000 examples, with 2,000 examples in each training task. For
 353 fine-tuning actors, we set epoch to 3 and learning rate to $5e-6$, and remains fixed during further
 354 training phase; for fine-tuning critics, we set epoch to 5 and learning rate to $5e-6$. We use the same
 355 base model for the actor and the critique model. For STaR and RL, we perform SFT to obtain an
 356 initialized model. In RL, we set KL coefficient to 0.01. In Critique-RL, we use RLOO as our base
 357 algorithm as it performs well and does not require a value model. In Stage II, β_1 is set to 0.2. We
 358 train the critique model for 500 steps at each stage and report best results. During evaluation, the
 359 temperature is set to 0. For inference-compute scaling and Pass@K, we set temperature to 0.7.

360 5.2 MAIN RESULTS

361 **Generally, critique models can significantly improve actor’s reasoning performance.** The re-
 362 sults in Table 1 demonstrate that when introducing critique models, the actor’s reasoning performance
 363 can be boosted by a large margin. For example, in the MATH task, even the SFT Baseline outperforms
 364 the model without a critic by 7.34 and 6.10 points on the 3B and 7B models, respectively. This
 365 suggests that critique models are an effective scalable oversight method, as discussed in Saunders
 366 et al. (2022); McAleese et al. (2024).

367 **RL-based methods outperforms fine-tuning-based ones.** Both SFT and STaR methods lead to
 368 promising critique models, but in most cases, online RL-based methods perform better, especially
 369 our Critique-RL. For instance, on the 3B model, our method surpasses the SFT method by an average
 370 of 7.11 points on accuracy across three datasets. It is worth noting that on AQuA, fine-tuning-based
 371 SFT and STaR may lead to negative impact on performance, while our method provides significant
 372 positive improvements. This reveals that online RL methods have greater potential and adaptability
 373 in eliciting the model’s critiquing ability, similar to the findings in McAleese et al. (2024).

374 **Critique-RL consistently outperforms other baselines in discrimination and final accuracy.** In
 375 terms of discrimination, our method also significantly outperforms other baselines, such as surpassing
 376 CTRL by 5.31, 6.36 points for 3B and 7B models on GSM8K, respectively. This reveals that our
 377 discrimination-related reward shaping can effectively optimizes discriminability. Thanks to this and
 378 the helpfulness reward design in the second stage, our method shows a significant improvement
 379 in final performance compared to other baselines. For example, on the 7B model, our method
 380 outperforms Retroformer by an average of 5.11 and 12.69 points on accuracy and discriminability,
 381 across three datasets.

378 5.3 ITERATIVE IMPROVEMENT OF CRITIQUE-RL
379

380 Furthermore, we validate the iterative improvement
381 capability of Critique-RL through two key
382 aspects: (1) Iterative refinement process: During
383 the i -th iteration, the critic generates critique
384 $c_i = \pi_\phi(x, y_0, c_1, \dots, c_{i-1}, y_{i-1})$, while
385 the actor produces the refined response $y_i =$
386 $\pi_\theta(x, y_0, c_1, \dots, y_{i-1}, c_i)$ accordingly. (2) Iterative
387 training process: We alternately conduct the
388 two-stage training of Critique-RL (Stage I and
389 Stage II) to optimize the critique model. The
390 detailed results are shown in Figure 4 and Table
391 2, respectively.

392 First, as demonstrated in Figure 4, through iterative
393 critique and refinement, the model exhibits
394 consistent Acc gains on Qwen2.5-3B, with each
395 iteration achieving measurable improvements.
396 Second, iterative training leads to further performance
397 enhancement, with detailed results using
398 Qwen2.5-3B on MATH dataset shown in Table
399 2. Specifically, both Stage I and Stage II
400 of Critique-RL demonstrate consistent improvement
401 in Acc and Acc@Dis metrics. Compared to the first iteration, the second iteration improves by
402 2.40 and 3.68 points on accuracy and discriminability.

403 6 DISCUSSION AND ANALYSIS

404 **Ablation on different stages.** We conduct ablation
405 experiments to validate the importance of
406 different components. The results are shown in
407 Table 3. Both Stage I and Stage II are crucial,
408 and removing either of them leads to a performance
409 drop. This indicates that optimizing both
410 discriminability and helpfulness is essential in
411 developing critique models.

412 **Ablation on reward design for Stage II.**
413 Next, we perform a deeper analysis of the
414 reward design in Stage II. First, if we remove
415 the discrimination-related r_{dis} and KL-based
416 regularization $\text{KL}(\pi_\phi^{\text{Stage-I}} \parallel \pi_\phi^{\text{Stage-II}})$, the discriminability
417 and accuracy suffer a significant drop. This further emphasizes that when optimizing for
418 helpfulness, it is crucial to maintain the model's discrimination ability. Second, when we replace
419 the reward function r_{refine} in Stage II with another reward function, i.e., r_Δ and $r_{\text{correction}}$, we observe
420 a slight performance drop. This may be because r_{refine} directly optimizes the Acc@Refine metric,
421 which aligns most closely with the test-time scenario.

422 **Analysis of helpfulness when the oracle verifier is available.** Many previous works have relied on
423 an external oracle verifier to assess the actor's reasoning results (Bai et al., 2022; Madaan et al., 2023;
424 Ye et al., 2023; Dhuliawala et al., 2024). In this scenario, the model's judgment ability is isolated,
425 allowing us to better evaluate the critique model's helpfulness. We conduct relevant experiments,
426 and the results are shown in Figure 5. We find that when the oracle verifier is available, all baselines
427 show performance improvements. In this case, our method still outperforms others across different
428 datasets and models, indicating that our approach significantly enhances the model's helpfulness.
429 Furthermore, comparisons with other RL baselines reveal that the optimization of discriminability in
430 our method also implicitly contributes to the improvement of helpfulness, suggesting that the two
431 abilities are not entirely independent. This further emphasizes the importance of optimizing both
432 abilities jointly in developing critique models.

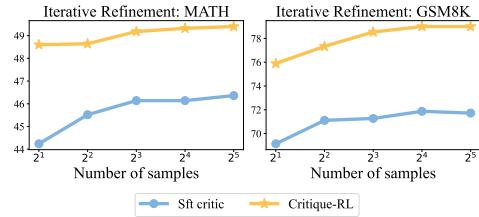


Figure 4: Results of iterative critique-refinement of Critique-RL using Qwen2.5-3B.

Table 2: Results of iterative training of Critique-RL using Qwen2.5-3B on MATH.

Method	Acc	Δ	Acc@Dis
No Critic	36.9	—	—
SFT	44.2	7.3	66.5
Critique-RL			
Iteration 1, Stage I	45.9	9.0	78.7
Iteration 1, Stage II	48.6	11.7	82.8
Iteration 2, Stage I	49.5	12.6	85.0
Iteration 2, Stage II	51.0	14.1	86.5

Table 3: Ablation study using Qwen2.5-3B. We report the Acc@Refine. “w/o” means without; “Stage II w/o discrimination” means in Stage II, we remove r_{dis} and $\text{KL}(\pi_\phi^{\text{Stage-I}} \parallel \pi_\phi^{\text{Stage-II}})$; “Stage II w/ r_Δ ” and “Stage II w/ $r_{\text{correction}}$ ” mean replacing the r_{refine} with the corresponding reward function.

Method	MATH		AQuA	
	Acc@Refine	Acc@Dis	Acc@Refine	Acc@Dis
Critique-RL (Ours)	48.6	82.8	56.7	69.9
-w/o Stage I	47.6	79.7	53.9	66.5
-w/o Stage II	45.9	78.7	54.7	68.2
-Stage II w/o discrimination	47.3	77.7	53.5	61.6
-Stage II w/ r_Δ	48.2	82.6	53.9	68.4
-Stage II w/ $r_{\text{correction}}$	47.7	82.0	54.7	68.4

This further emphasizes that when optimizing for helpfulness, it is crucial to maintain the model's discrimination ability. Second, when we replace the reward function r_{refine} in Stage II with another reward function, i.e., r_Δ and $r_{\text{correction}}$, we observe a slight performance drop. This may be because r_{refine} directly optimizes the Acc@Refine metric, which aligns most closely with the test-time scenario.

Many previous works have relied on an external oracle verifier to assess the actor's reasoning results (Bai et al., 2022; Madaan et al., 2023; Ye et al., 2023; Dhuliawala et al., 2024). In this scenario, the model's judgment ability is isolated, allowing us to better evaluate the critique model's helpfulness. We conduct relevant experiments, and the results are shown in Figure 5. We find that when the oracle verifier is available, all baselines show performance improvements. In this case, our method still outperforms others across different datasets and models, indicating that our approach significantly enhances the model's helpfulness. Furthermore, comparisons with other RL baselines reveal that the optimization of discriminability in our method also implicitly contributes to the improvement of helpfulness, suggesting that the two abilities are not entirely independent. This further emphasizes the importance of optimizing both abilities jointly in developing critique models.

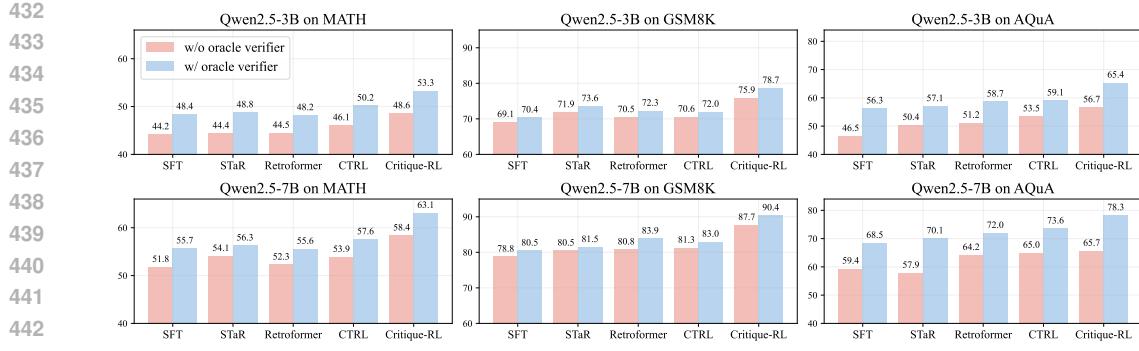


Figure 5: Performance with and without the oracle verifier. When the oracle verifier is available, the model no longer needs to make discrimination and just needs to provide useful feedback. This allows us to evaluate the model’s helpfulness more accurately.

Evaluation of test-time inference compute scaling for Critique-RL. We investigate whether Critique-RL can be combined with inference-time compute scaling strategy. Following Qu et al. (2024); Snell et al. (2024); Xi et al. (2024), we leverage the commonly used majority vote (MV@ K) (Wang et al., 2023) which evaluates whether the most frequent answer among K samples is correct. The results of MATH are shown in Figure 1 and the results of GSM8K are shown in Figure 6 of Appendix F. Compared to the baseline, Critique-RL significantly increases the performance ceiling and shows a more sustained upward trend as inference compute scales. More importantly, performing $K \times$ response-critique-refinement sampling is more compute-efficient than conducting $3K \times$ parallel sampling responses, suggesting the compute-efficiency of Critique-RL.

Generalization to OOD tasks. We also validate the generalization of the models trained by Critique-RL on OOD tasks. The results in Table 4 show that the models trained still deliver significant performance improvements, further demonstrating the potential of this scalable oversight approach.

Performance of our trained critique models on actor models with different capability levels. To further investigate the Critique-RL in varying base models, we conduct two types of experiments. In the first setting, we use a strong reasoning model DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025) as our actor model while using Qwen2.5-7B as our critic model. This evaluation setting investigates the generalization of Critique-RL to reasoning models. The

results in Table 5 reveal that, besides non-reasoning models (Qwen2.5-3B, Qwen2.5-7B) with structured CoT, our method is also effective for reasoning models with complex CoT structures on both in-domain and out-of-domain tasks, particularly in terms of the Acc@Dis achieved by the critique models. While DeepSeek-R1-Distill-Qwen-7B already performs strongly on MATH-500, critique models can still offer marginal gains in reasoning accuracy. More impressively, on the TheoremQA dataset which spans diverse domains including Math, EECS, Physics and Finance, critique models substantially boost performance, highlighting the strong generalization ability of our approach. Notably, Critique-RL outperforms SFT, Retroformer, and CTRL by 26.75, 28.75, 29.88 points in Acc@Dis, respectively, on the TheoremQA dataset—doubling the performance of these baselines.

In the second setting, we use Qwen2.5-72B-Instruct as the actor model and Qwen2.5-7B as the critique model to investigate weak-to-strong generalization.

Table 4: Out-of-domain evaluation of Critique-RL.

Model	Method	SVAMP		TheoremQA	
		Acc	Pass@10	Acc	Pass@10
Qwen2.5-3B	No Critic	70.7	92.0	15.1	34.8
	SFT	74.7	95.7	15.3	36.1
	Retroformer	75.0	96.0	16.1	37.0
	CTRL	76.0	95.7	15.8	36.5
	Critique-RL	78.3	96.3	16.8	37.8
Qwen2.5-7B	No Critic	80.3	95.7	19.4	39.8
	SFT	83.0	95.7	20.5	41.9
	Retroformer	84.0	96.0	20.0	42.3
	CTRL	85.1	96.7	21.1	42.9
	Critique-RL	89.7	97.0	21.4	43.0

Table 5: Performance on DeepSeek-R1-Distill-Qwen-7B as actor.

Method	In-Domain: MATH-500			OOD: TheoremQA		
	Acc	Δ	Acc@Dis	Acc	Δ	Acc@Dis
No Critic	84.60	-	-	21.63	-	-
SFT	85.60	1.00	83.40	29.75	8.13	24.38
Retroformer	85.80	1.20	84.80	29.38	7.75	22.38
CTRL	85.80	1.20	84.80	29.00	7.38	21.25
Critique-RL	86.60	2.00	93.00	30.38	8.75	51.13

486 The results in Table 6 show that Critique-
 487 RL improves actor performance even in
 488 large-scale settings, though with less pro-
 489 nounced gains compared to smaller-actor
 490 settings. Nonetheless, it still outperforms
 491 baselines on both in-domain and out-
 492 of-domain tasks. Notably, our method
 493 achieves significantly higher discrimina-
 494 tion, confirming the effectiveness of our discrimination-based reward shaping.

495 **More experiments and qualitative analysis.** We conduct extensive experiments to show the effec-
 496 tiveness and working mechanism of Critique-RL, with the detailed results presented in the Appendix:
 497 (1) In addition to the Qwen2.5 series (Team, 2024), we evaluate our method on different architectures
 498 including Llama3.2 (see Appendix D). (2) We compare Critique-RL with other refinement methods
 499 including Self-Refine (Madaan et al., 2023), SuperCorrect (Yang et al., 2024) and Critic-Cot (Zheng
 500 et al., 2024), and the results are presented in Appendix E. (3) We also perform test-time scaling
 501 analysis of sampling multiple refinement on the same response, with results presented in Appendix
 502 F. (4) We conduct experiments on summarization tasks using CNN/DailyMail (Hermann et al.,
 503 2015) dataset to investigate our method’s generalization ability on open-ended tasks where rule-based
 504 verifier cannot be directly applied, the results are in Appendix G. (5) We perform a qualitative analysis
 505 on how Critique-RL works and provide several examples in Appendix J.

506 7 CONCLUSION

507 In this paper, we propose Critique-RL, an RL approach for developing critique models. Through
 508 in-depth analysis, we highlight the importance of explicitly optimizing model discriminability and
 509 propose a two-stage RL approach that effectively optimizes both discriminability and helpfulness.
 510 We validate its stability and superiority through detailed experiments, and further uncover its working
 511 mechanism through ablation studies and analyses. We hope that our work can provide insights for the
 512 scalable oversight community of language models.

513 ETHICS STATEMENT

514 This paper presents Critique-RL, a novel two-stage RL approach to develop critiquing language
 515 models for providing accurate assessment and helpful feedback for model outputs. We firmly state
 516 that this work is intended for ethical and constructive purpose. While no immediate societal harms
 517 are evident, this approach enables scalable supervision by training models with minimal direct human
 518 oversight. Nevertheless, its potential susceptibility to misuse warrants proactive measures to ensure
 519 responsible governance.

520 REPRODUCIBILITY STATEMENT

521 We claim our detailed experiment setting in §5.1. In addition, we upload anonymized versions of our
 522 data and code in a Zip file with a Readme file to ensure easy reproduction of all reported results.

523 REFERENCES

524 Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin,
 525 Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for
 526 learning from human feedback in llms. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar
 527 (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*
 528 (*Volume 1: Long Papers*), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pp. 12248–12267.
 529 Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.662. URL
 530 <https://doi.org/10.18653/v1/2024.acl-long.662>.

531 AIME. Aime problems and solution, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

532 Table 6: Performance on Qwen2.5-72B-Instruct as actor.

Method	In-Domain: MATH-500			OOD: TheoremQA		
	Acc	Δ	Acc@Dis	Acc	Δ	Acc@Dis
No Critic	79.10	-	-	21.38	-	-
SFT	79.20	0.10	80.20	21.63	0.25	23.00
Retroformer	79.20	0.10	80.60	21.75	0.38	21.38
CTRL	79.40	0.30	79.40	21.50	0.13	21.13
Critique-RL	80.30	1.20	89.20	23.50	2.10	46.63

540 Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket
 541 Tandon. RL4F: generating natural language feedback with reinforcement learning for repairing
 542 model outputs. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of
 543 the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),
 544 ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 7716–7733. Association for Computational
 545 Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.427.

546 Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna
 547 Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness
 548 from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

549

550 Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile
 551 Lukosiute, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron
 552 McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-
 553 Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal
 554 Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova
 555 DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec,
 556 Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan
 557 Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. Measuring progress on
 558 scalable oversight for large language models. *CoRR*, abs/2211.03540, 2022. doi: 10.48550/ARXIV.
 559 2211.03540.

560 Wenhui Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang,
 561 and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In Houda Bouamor,
 562 Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in
 563 Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 7889–7901.
 564 Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.489.
 565 URL <https://doi.org/10.18653/v1/2023.emnlp-main.489>.

566 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 567 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 568 Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

569 DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning,
 570 2025. URL <https://arxiv.org/abs/2501.12948>.

571

572 Shehzaad Dhulaiwala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and
 573 Jason Weston. Chain-of-verification reduces hallucination in large language models. In Lun-Wei
 574 Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational
 575 Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 3563–3578.
 576 Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.212.

577 Yiwen Ding, Zhiheng Xi, Wei He, Lizhuoyuan Lizhuoyuan, Yitao Zhai, Shi Xiaowei, Xunliang Cai,
 578 Tao Gui, Qi Zhang, and Xuanjing Huang. Mitigating tail narrowing in LLM self-improvement via
 579 socratic-guided sampling. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the
 580 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Lin-
 581 guistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10627–10646, Albuquerque,
 582 New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6.
 583 URL <https://aclanthology.org/2025.naacl-long.533/>.

584 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
 585 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn,
 586 Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston
 587 Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron,
 588 Bin Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris
 589 McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton
 590 Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David
 591 Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes,
 592 Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip
 593 Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme
 Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu,

594 Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan
 595 Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vrane, Jason Park, Jay Mahadeokar, Jeet
 596 Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi,
 597 Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph
 598 Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani,
 599 Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*,
 600 abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783.

601 Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen.
 602 CRITIC: large language models can self-correct with tool-interactive critiquing. In *The Twelfth*
 603 *International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11,*
 604 2024. OpenReview.net, 2024.

605 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang,
 606 Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with
 607 the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings*
 608 *of the Neural Information Processing Systems Track on Datasets and Benchmarks*
 609 *1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL
 610 [https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/
 611 hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html](https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html).

612 Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa
 613 Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In Corinna
 614 Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.),
 615 *Advances in Neural Information Processing Systems 28: Annual Conference on Neural In-*
 616 *formation Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp.
 617 1693–1701, 2015. URL [https://proceedings.neurips.cc/paper/2015/hash/
 618 afdec7005cc9f14302cd0474fd0f3c96-Abstract.html](https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html).

619 Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song,
 620 and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth*
 621 *International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11,*
 622 2024. OpenReview.net, 2024.

623 Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot,
 624 Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,
 625 Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas
 626 Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi:
 627 10.48550/ARXIV.2310.06825.

628 Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli,
 629 Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava,
 630 Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust.
 631 Training language models to self-correct via reinforcement learning. *CoRR*, abs/2409.12917, 2024.
 632 doi: 10.48550/ARXIV.2409.12917.

633 Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale gen-
 634 eration: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-
 635 Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational*
 636 *Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp.
 637 158–167. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1015. URL
 638 <https://doi.org/10.18653/v1/P17-1015>.

639 Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen,
 640 Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference
 641 data curation via human-ai synergy. *CoRR*, abs/2507.01352, 2025. doi: 10.48550/ARXIV.2507.
 642 01352. URL <https://doi.org/10.48550/arXiv.2507.01352>.

643 Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhui Chen. General-reasoner:
 644 Advancing llm reasoning across all domains. *arXiv:2505.14652*, 2025. URL <https://arxiv.org/abs/2505.14652>.

648 Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon,
 649 Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder,
 650 Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative
 651 refinement with self-feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko,
 652 Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

653
 654 Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz,
 655 and Jan Leike. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*, 2024.

656
 657 OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774.

658
 659 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong
 660 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton,
 661 Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and
 662 Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi
 663 Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

664
 665 Arkil Patel, Satwik Bhattacharya, and Navin Goyal. Are NLP models really able to solve simple math
 666 word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür,
 667 Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational
 668 Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 2080–
 669 2094. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.
 670 168. URL <https://doi.org/10.18653/v1/2021.naacl-main.168>.

671
 672 Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching
 673 language model agents how to self-improve. *CoRR*, abs/2407.18219, 2024. doi: 10.48550/ARXIV.
 674 2407.18219.

675
 676 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien
 677 Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a
 678 benchmark. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=Ti67584b98>.

679
 680 William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan
 681 Leike. Self-critiquing models for assisting human evaluators. *CoRR*, abs/2206.05802, 2022. doi:
 682 10.48550/ARXIV.2206.05802.

683
 684 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li,
 685 Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open
 686 language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL
 687 <https://doi.org/10.48550/arXiv.2402.03300>.

688
 689 Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflex-
 690 ion: language agents with verbal reinforcement learning. In Alice Oh, Tristan Naumann, Amir
 691 Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information
 692 Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS
 693 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

694
 695 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally
 696 can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024. doi: 10.48550/
 697 ARXIV.2408.03314.

698
 699 Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient
 700 methods for reinforcement learning with function approximation. In Sara A. Solla, Todd K. Leen,
 701 and Klaus-Robert Müller (eds.), *Advances in Neural Information Processing Systems 12, [NIPS
 Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pp. 1057–1063. The MIT
 Press, 1999.

702 Zhengyang Tang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei
 703 Huang, Tianyu Liu, Bowen Yu, et al. Enabling scalable oversight via self-evolving critic. *arXiv*
 704 *preprint arXiv:2501.05727*, 2025.

705 706 Qwen Team. Qwen2.5: A party of foundation models, September 2024.

707 Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay
 708 Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian
 709 Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esibou, Jude Fernandes, Jeremy Fu, Wenyin
 710 Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar
 711 Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann,
 712 Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana
 713 Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor
 714 Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan
 715 Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang,
 716 Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang,
 717 Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey
 718 Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*,
 719 abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288.

720 Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin,
 721 Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun
 722 Zhao, Xiao Wang, Tao Ji, Hang Yan, Lixing Shen, Zhan Chen, Tao Gui, Qi Zhang, Xipeng Qiu,
 723 Xuanjing Huang, Zuxuan Wu, and Yu-Gang Jiang. Secrets of RLHF in large language models part
 724 II: reward modeling. *CoRR*, abs/2401.06080, 2024. doi: 10.48550/ARXIV.2401.06080.

725 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha
 726 Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language
 727 models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

728 729 Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin
 730 Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference
 731 on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

732 Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang
 733 Hong, Shihan Dou, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai,
 734 Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu,
 735 Xuanjing Huang, and Yu-Gang Jiang. Enhancing LLM reasoning via critique models with test-time
 736 and training-time supervision. *CoRR*, abs/2411.16579, 2024. doi: 10.48550/ARXIV.2411.16579.
 737 URL <https://doi.org/10.48550/arXiv.2411.16579>.

738 Zhihui Xie, Jie Chen, Liyu Chen, Weichao Mao, Jingjing Xu, and Lingpeng Kong. Teaching language
 739 models to critique via reinforcement learning. *CoRR*, abs/2502.03492, 2025. doi: 10.48550/
 740 ARXIV.2502.03492. URL <https://doi.org/10.48550/arXiv.2502.03492>.

741 742 Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride
 743 and prejudice: LLM amplifies self-bias in self-refinement. In Lun-Wei Ku, Andre Martins,
 744 and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for
 745 Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16,
 746 2024, pp. 15474–15492. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.
 747 ACL-LONG.826.

748 Ling Yang, Zhaochen Yu, Tianjun Zhang, Minkai Xu, Joseph E. Gonzalez, Bin Cui, and Shuicheng
 749 Yan. Supercorrect: Supervising and correcting language models with error-driven insights. *CoRR*,
 750 abs/2410.09008, 2024. doi: 10.48550/ARXIV.2410.09008. URL <https://doi.org/10.48550/arXiv.2410.09008>.

752 Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh R. N.,
 753 Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong,
 754 and Silvio Savarese. Retroformer: Retrospective large language agents with policy gradient
 755 optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

756 Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeyonbin Hwang, and Minjoon Seo.
757 Selfee: Iterative self-revising llm empowered by self-feedback generation. Blog post, May 2023.
758

759 Siyu Yuan, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. Agent-r: Training
760 language model agents to reflect via iterative self-training. *arXiv preprint arXiv:2501.11425*, 2025.

761 Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with
762 reasoning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh
763 (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural
764 Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 -
765 December 9, 2022*, 2022.

766 Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin,
767 Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu,
768 Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun,
769 Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of RLHF in large
770 language models part I: PPO. *CoRR*, abs/2307.04964, 2023. doi: 10.48550/ARXIV.2307.04964.
771

772 Xin Zheng, Jie Lou, Boxi Cao, Xueru Wen, Yuqiu Ji, Hongyu Lin, Yaojie Lu, Xianpei Han, Debing
773 Zhang, and Le Sun. Critic-cot: Boosting the reasoning abilities of large language model via
774 chain-of-thoughts critic, 2024.

775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

810 A THE USE OF LARGE LANGUAGE MODELS
811812 LLMs are utilized in this manuscript for partial grammatical checks and language polishing. The
813 authors are fully responsible for the final content.
814815 B ALGORITHM OF CRITIQUE-RL
816817 Our main algorithm is summarized in Algorithm 1.
818820 **Algorithm 1:** Critique-RL821 **Input:** Actor model π_θ , base critique model π_ϕ , SFT dataset \mathcal{D}_{SFT} , RL dataset \mathcal{D}_{RL} , function that
822 extracts the correctness of a response judged by a critique f , oracle reward function
823 r_{oracle} , discrimination reward function r_{dis} .
824824 **Procedure** Supervised Fine-tuning:825 $\pi_\phi^{\text{SFT}} \leftarrow \pi_\phi$;
826 Update π_ϕ^{SFT} by minimizing $\mathcal{L}_{\text{SFT}}(\phi) = \mathbb{E}_{(x,y,c) \sim \mathcal{D}_{\text{SFT}}} [\log \pi_\phi(c|x, y)]$;
827828 **Procedure** Critique-RL Stage I: optimizing discriminability through direct reward signals.
829829 $\pi_\phi^{\text{Stage-I}} \leftarrow \pi_\phi^{\text{SFT}}$;
830 **for** batch in \mathcal{D}_{RL} **do**
831 **for** x in batch **do**
832 Generate y and c with π_θ and $\pi_\phi^{\text{Stage-I}}$;
833 Compute discrimination reward with $r_{\text{dis}}(x, y, c) = \mathbb{1}(f(x, y, c) = r_{\text{oracle}}(x, y))$;
834 **end**
835 Update $\pi_\phi^{\text{Stage-I}}$ by maximizing
836 $\mathbb{E}_{c \sim \pi_\phi^{\text{Stage-I}}(\cdot|x, y)} [r_{\text{dis}}(x, y, c) - \beta \text{KL}(\pi_\phi^{\text{SFT}}(c|x, y) || \pi_\phi^{\text{Stage-I}}(c|x, y))]$;
837 **end**
838839 **Procedure** Critique-RL Stage II: optimization helpfulness while maintaining discriminability.
840840 $\pi_\phi^{\text{Stage-II}} \leftarrow \pi_\phi^{\text{Stage-I}}$;
841 **for** batch in \mathcal{D}_{RL} **do**
842 **for** x in batch **do**
843 Generate y, c and y' with π_θ and $\pi_\phi^{\text{Stage-II}}$;
844 Compute discrimination reward with $r_{\text{dis}}(x, y, c) = \mathbb{1}(f(x, y, c) = r_{\text{oracle}}(x, y))$;
845 Compute refinement reward with $r_{\text{refine}} = r_{\text{oracle}}(x, y')$;
846 **end**
847 Update $\pi_\phi^{\text{Stage-II}}$ by maximizing $\mathbb{E}_{c \sim \pi_\phi^{\text{Stage-II}}(\cdot|x, y), y' \sim \pi_\theta(\cdot|x, y, c)} [r_{\text{refine}} + \beta_1 r_{\text{dis}}(x, y, c) -$
848 $\beta_2 \text{KL}(\pi_\phi^{\text{Stage-I}}(c|x, y) || \pi_\phi^{\text{Stage-II}}(c|x, y))]$.
849 **end**
850853 C PERFORMANCE ON MORE CHALLENGING BENCHMARKS
854855
856
857 To further validate the effectiveness of Critique-RL, we conduct experiments using Qwen2.5-7B-
858 Instruct and evaluated on the AIME2024, AIME2025 (AIME, 2025), and GPQA-Diamond (Rein et al.,
859 2024) benchmarks. We used General-Reasoner-7B (Ma et al., 2025) as the actor and constructed
860 a training set of 30k examples based on the WebInstruct-Verified (Ma et al., 2025) dataset for
861 RL training. The experimental results are in Table 7. The experimental results demonstrate that
862 our method significantly improves the critique model’s discriminability, with particularly notable
863

Table 7: Performance on challenging benchmarks using Qwen2.5-7B-Instruct.

Methods	GPQA-Diamond			AIME2024			AIME2025		
	Acc	Δ	Acc@Dis	Acc	Delta	Acc@Dis	Acc	Δ	Acc@Dis
No Critic	34.30	-	-	11.98	-	-	6.67	-	-
SFT	35.86	1.56	41.20	12.30	0.32	22.71	7.50	0.83	6.67
Critique-RL	37.37	3.07	51.52	13.75	1.77	53.44	8.50	1.83	30.10

improvements on the challenging reasoning datasets AIME2024 and AIME2025. Without fine-grained hyperparameter adjustments, our method outperforms the baseline across all three benchmarks, proving the effectiveness of Critique-RL in complex reasoning scenarios.

D PERFORMANCE ON VARYING MODEL SERIES

Table 8: Performance on Llama3.2-3B with GSM8K.

Method	GSM8K		
	Acc	Δ	Acc@Dis
No Critic	49.28	-	-
SFT	50.80	1.52	68.11
Retroformer	52.08	2.81	63.85
CTRL	52.24	2.96	66.01
Critique-RL	52.99	3.72	75.04

To evaluate the effectiveness and generalization capability of Critique-RL, we conduct experiments using the Llama3.2-3B (Dubey et al., 2024) model on the GSM8K dataset. As shown in Table 8, Critique-RL proves effective not only on Qwen2.5 models but also on Llama3.2 models, particularly in enhancing the discriminability of the critique models. These results highlight the adaptability and robust performance of Critique-RL across different model architectures.

E COMPARISON WITH OTHER IMPORTANT REFINEMENT METHODS

To further validate the advantages of Critique-RL over other refinement methods, we conduct evaluations of other refinement methods including Self-Refine (Madaan et al., 2023), SuperCorrect (Yang et al., 2024) and Critic-CoT (Zheng et al., 2024) with Qwen2.5-3B on GSM8K. For a fairer comparison, we train the models in Self-Refine and Critic-CoT using the same dataset (sampled from Qwen2.5-3B-Instruct) as Critique-RL. In terms of SuperCorrect, we choose Deepseek-R1 (DeepSeek-AI, 2025) as the teacher model to create both the Hierarchical Thought Templates and positive critique datasets. The results are presented in Table 9. Critique-RL significantly outperforms all other methods in both Acc and Acc@Dis, surpassing Critic-CoT and SuperCorrect by 5.31 and 3.11 points in terms of Acc, respectively. Moreover, Critique-RL outperforms Self-Refine across refinement iterations, demonstrating its greater effectiveness. Notably, SuperCorrect exhibited poor discriminability, likely because it simply used teacher model data as positive examples and student model data as negative ones for DPO training. Given the GSM8K dataset’s simplicity, the student model’s output is not consistently inferior to teacher model’s, leading to potential impairment to the model’s discriminability.

These refinement methods are implemented using SFT (Self-Refine), self-improve (Critic-CoT) or intricate SFT+DPO (SuperCorrect) approaches, whereas Critique-RL employs an online RL methodology, which accounts for its observed performance advantages.

F MORE TEST-TIME SCALING RESULTS

The results of inference compute scaling on GSM8K are illustrated in Figure 6. Similar to the findings on MATH, Critique-RL is more compute-efficient and significantly increases the performance ceiling, validating the potential of our approach. In addition, we evaluate the refine compute scaling of SFT and Critique-RL across MATH, GSM8K, and AQUA, as illustrated in Figure 7. Critique-RL

918 Table 9: Comparison with other refinement methods with Qwen2.5-3B on GSM8K.
919

Method	GSM8K	
	Acc	Acc@Dis
Self-Refine iteration=1	71.42	75.84
Self-Refine iteration=2	72.71	76.52
Critic-CoT	70.58	74.70
SuperCorrect	72.78	62.17
Critique-RL (Ours)	75.89	87.44

920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
consistently achieves approximately twice the sampling efficiency of SFT. Notably, with the 7B
model on GSM8K, Critique-RL's Pass@1 even surpasses the SFT's Pass@64, demonstrating the
effectiveness of our approach.

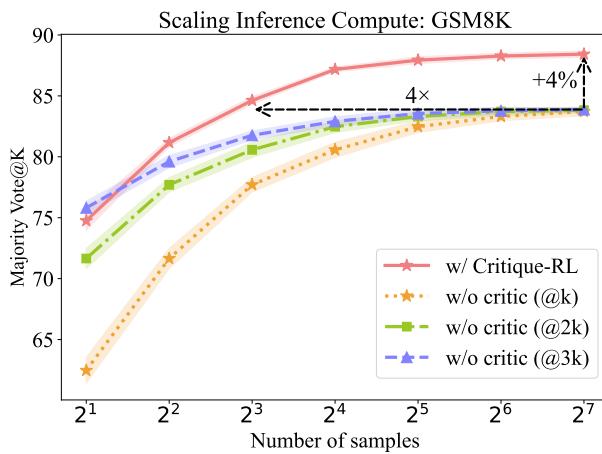


Figure 6: Inference compute scaling for Critique-RL, with @2k and @3k indicating sampling amounts that are 2 times and 3 times the x-axis value, respectively. Critique-RL improves the performance ceiling and is more compute-efficient.

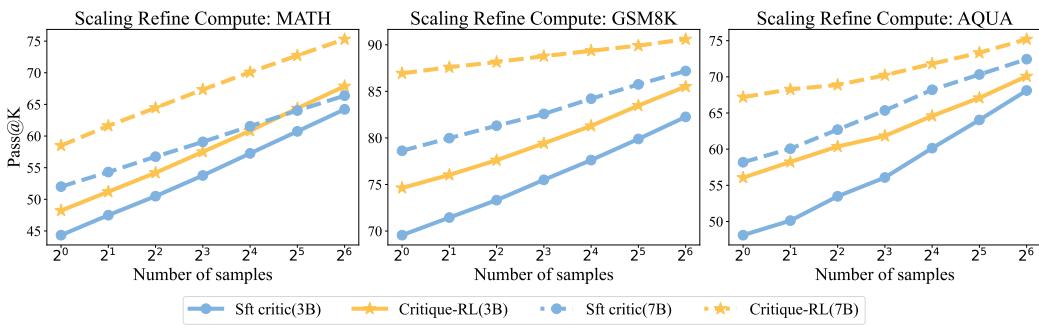


Figure 7: Refine compute scaling for Critique-RL and SFT critic with Qwen2.5-3B and Qwen2.5-7B.

G PERFORMANCE ON SUMMARIZATION TASK

For open-ended tasks where rule-based verifiers cannot be directly applied, reward signals can be provided through additional reward models or AI feedback (e.g., using GPT-4o (OpenAI, 2023) for judgement).

We conduct experiments of Critique-RL with Qwen2.5-7B-Instruct (Team, 2024) on summarization task using CNN/DailyMail (Hermann et al., 2015) dataset. Specifically, given an article x , the actor model generates an original summary y . The reward model (Skywork-Reward-V2-Llama-3.1-8B (Liu et al., 2025)) then evaluates the summary, with its output linearly scaled to a 1-10 range, i.e., $r_{\text{oracle}}(x, y)$. Subsequently, the critique model produces critique c , which includes comments about the summary across key criteria, a quality score from 1-10, and improvement suggestions. The actor model then generates a revised summary y' accordingly, which is also scored by the reward model to yield a refinement score $r_{\text{refine}} = r_{\text{oracle}}(x, y')$. Based on this, we define the discrimination reward function of the critique model as:

$$r_{\text{dis}}(x, y, c) = \max(0, 1 - \frac{|f(x, y, c) - r_{\text{oracle}}(x, y)|}{\delta})$$

where $f(x, y, c)$ is the quality score of the original summary from critique model. δ is the permissible maximum error range.

In stage I, we optimize the discriminability of the critique model using $r_{\text{dis}}(x, y, c)$; In stage II, we optimize the helpfulness while maintaining discriminability using the following reward function:

$$r_{\text{stageII}} = r_{\text{refine}} + \beta_1 r_{\text{dis}}(x, y, c)$$

In our experiments, we select 5000 training and 1000 test queries from CNN/DailyMail 3.0.0's official splits. The results are presented in the Table 10.

The results reveal that Critique-RL can effectively optimize discriminability, yielding improvement in summary quality. We use MSE and MAE to measure the error between the quality scores produced by the critique model and those from the reward model. Specifically, Critique-RL outperforms baseline by 0.87 points in Score, 7.87 points in MSE@Dis and 1.79 points in MAE@Dis. These improvements demonstrate the strong generalization ability of our approach to open-ended tasks, contributing to scalable oversight.

Table 10: Performance on summarization task using Qwen2.5-7B-Instruct. We report the original Score by reward model. The MSE@Dis stands for mean square error, and MAE@Dis stands for mean absolute error, where smaller values indicate stronger discrimination abilities.

Method	CNN/MD			
	Score↑	Delta↑	MSE@Dis↓	MAE@Dis↓
No Critic	19.69	-	-	-
7B-Instruct	19.94	0.25	9.46	2.77
Critique-RL (Ours)	20.81	1.12	1.59	0.98

H VALIDATING THE EFFECTIVENESS OF CRITIQUE MODEL

Introducing a separate critique model leads to increased manual effort and additional complexity. To validate the usage of the critique model, we compare Critique-RL with actor-only RL method to show that training a critique model provides significant benefits over directly optimizing the actor. In particular, for actor-only method, we conduct experiments on directly RL the actor and SCoRe (Kumar et al., 2024); for actor-critic paradigm, we use a SFT-based critique model as well as our Critique-RL. For a fairer comparison, we train the actor model using the same reasoning traces as Critique-RL in direct RL and using the same reasoning, critique and refinement dataset as Critique-RL in SCoRe. All experiments are conducted with Qwen2.5-7B on the Math dataset.

Table 11: Comparison with actor-only RL method.

Category	Method	MATH	
		Acc	Acc@Dis
Actor-only	Directly RL	49.78	-
	SCoRe	56.52	72.51
	SFT	51.84	67.59
Actor-Critique	Critique-RL	58.40	85.20

The results in Table 11 show that Critique-RL significantly outperforms Directly RL by 8.62 points in terms of Acc. Also Critique-RL outperforms SCoRe by 12.69 points in terms of Acc@Dis, and 1.88 points in terms of Acc. Note that during the training process of Critique-RL, the actor model

1026 remained fixed and is thus inherently weaker in reasoning and refinement than the trained SCoRe
 1027 actor model. Importantly, the trained critique model can be flexibly applied to other stronger actor
 1028 models (weak-to-strong) and reasoning models to further improve their performance (see Section 6).
 1029 This modularity and transferability are advantages that SCoRe lacks.

1030 Moreover, we conduct the test-time scaling experiment. The majority vote (MV@K) results are as
 1031 shown in Table 12. The results show that even the actor model has been well-trained, generating
 1032 parallel responses still underperforms Critique-RL’s response-critique-refinement process. Notably,
 1033 Critique-RL’s MV@1 even surpasses Directly RL’s MV@12. This highlights the compute-efficiency
 1034 of Critique-RL.
 1035

1036 Table 12: Performance comparison between Directly RL and Critique-RL under MV@K.
 1037

K	Directly RL			Critique-RL MV@K
	MV@K	MV@2K	MV@3K	
1	49.78	50.05	52.39	58.40
2	50.05	53.49	55.04	59.10
4	53.49	55.08	56.75	65.91

I SENSITIVITY ANALYSIS

For solidness, we provide details about different values for β , β_1 , β_2 and training steps per stage.

Experiments on different values for β , β_1 , and β_2 . We exemplify our selection of the parameters β , β_1 , and β_2 by presenting the performance of the Qwen2.5-3B model on the GSM8K dataset as an example. The results in Table 13 reveal that these parameters are not sensitive, so we ultimately choose $\beta = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.95$ for our experiments.

Experiments on different training steps per stage. We show the performance of the two stages of Critique-RL at different training steps with Qwen2.5-3B on MATH dataset. The results in Table 14 indicate that within 500 steps of Stage I, the model’s discriminability was substantially enhanced, with Acc@Dis rising from 66.51 to 78.68. During Stage II, the model maintained this discriminability while further improving helpfulness, with Acc increasing from 45.90 to 48.60.

While further refinement of parameters could potentially yield additional performance gains, the current experimental outcomes are already statistically sound and adequately substantiate our core conclusions.

J QUALITATIVE ANALYSIS

We perform a qualitative investigation into how Critique-RL works and provide several examples in Appendix J. In Figure 8, facing the originally incorrect response, the critique model after SFT is unable to detect errors, leading the actor’s refinement response to retain the same errors. However, the model trained after Critique-RL identifies the errors in the original response and provides detailed, constructive suggestions for modification, leading to the correct refinement response. In Figure 9, model trained after Critique-RL Stage I is able to detect errors, demonstrating its discriminability. However, the model provides the actor with low-quality suggestion, causing the actor’s refinement response to be incorrect. In contrast, for the same erroneous original response, model trained after Critique-RL Stage II not only detects the error but also offers a constructive suggestion, ultimately leading to the correct refinement response, demonstrating the advantage of two-stage RL process.

To directly assess the quality of critiques generated by Critique-RL, we randomly collect 600 critiques that successfully helped refine incorrect answer into correct ones. We leverage GPT-4o with ground-truth answers and solutions as references to evaluate quality more accurately. The results show that 96.2% of these critiques made correct discriminative judgments, and 93.3% were rated as high-quality, demonstrating that Critique-RL produces reliable and helpful critiques.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089

Table 13: Results of different values for β , β_1 , and β_2 with Qwen2.5-3B on GSM8K.

Parameter	Value	Acc	Delta	Acc@Dis
β	0.008	74.60	8.57	86.24
	0.01	75.89	9.86	87.44
	0.012	74.22	8.19	87.10
β_1	0.88	74.60	8.57	86.18
	0.9	75.89	9.86	87.44
	0.92	74.68	8.65	86.09
β_2	0.93	74.68	8.65	85.99
	0.95	75.89	9.86	87.44
	0.97	74.37	8.34	85.74

1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117

Table 14: Results of different training steps per stage with Qwen2.5-3B on MATH.

Step	Critique-RL Stage I		Critique-RL Stage II	
	Acc	Acc@Dis	Acc	Acc@Dis
0	44.24	66.51	45.90	78.68
100	44.22	68.26	45.88	80.56
200	44.60	71.53	46.82	81.77
300	44.89	75.72	47.02	82.47
400	45.18	78.20	47.90	83.06
500	45.90	78.68	48.60	82.80

1126
1127
1128
1129
1130
1131
1132
1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

<p>Problem: Each person in a certain household consumes 0.2 kg of rice every meal. Supposing 5 members of the household eat rice every lunch and dinner, how many weeks will a 42 kg bag of rice last?</p>	
<p>Ground Truth Answer:</p> <ul style="list-style-type: none"> Each person eats 0.2 kg/meal \times 2 meals/day = <<0.2*2=0.4>>0.4 kg of rice per day. So, the 5 members consume 0.4 kg/day/person \times 5 people = <<0.4*5=2>>2 kg of rice per day. In a week, the household consumes 2 kg/day \times 7 days/week = <<2*7=14>>14kg of rice. Thus, 42 kg of rice will last 42 kg / 14 kg/week = <<42/14=3>>3 weeks. <p>The answer is 3.</p>	<p>Actor Model (original response):</p> <p>Let's break it down step by step:</p> <ol style="list-style-type: none"> The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day. The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week. A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks. <p>The answer is 21.</p> <p>Correctness Verifier: </p>
<p>Critique Model after SFT</p> <p>Critique Model (critique):</p> <p>Step sentence: The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day.</p> <p>Correctness of the step: Correct</p> <p>Explanation: This step correctly calculates the daily rice consumption by multiplying the amount consumed per meal (0.2 kg) by the number of meals per day (2).</p> <p>Step sentence: The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week.</p> <p>Correctness of the step: Correct</p> <p>Explanation: This step correctly calculates the weekly rice consumption by multiplying the daily consumption (0.4 kg) by the number of days in a week (5).</p> <p>Step sentence: A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks.</p> <p>Correctness of the step: Correct</p> <p>Explanation: This step correctly calculates the number of weeks the 42 kg bag of rice will last by dividing the total amount of rice by the weekly consumption (2 kg per week).</p> <p>Correctness of the final answer: Correct</p> <p>Explanation: The final answer of 21 weeks is correct based on the calculations provided in the solution.</p> <p>Discrimination Verifier: </p>	<p>Critique Model after Critique-RL Training</p> <p>Critique Model (critique):</p> <p>Step sentence: The household consumes 0.2 kg of rice per meal * 2 meals per day = 0.4 kg of rice per day.</p> <p>Correctness of the step: Wrong</p> <p>Explanation: This step is wrong because it incorrectly calculates the daily rice consumption. Since 5 members of the household eat rice every lunch and dinner, and each person consumes 0.2 kg of rice per meal, the daily consumption is: 5 members \times 2 meals/day \times 0.2 kg/meal = 2 kg/day</p> <p>Step sentence: The household consumes 0.4 kg of rice per day * 5 days per week = 2 kg of rice per week.</p> <p>Correctness of the step: Wrong</p> <p>Explanation: This step is wrong because it incorrectly calculates the weekly rice consumption. Given the correct daily consumption is 2 kg/day, the weekly consumption is: 2 kg/day \times 7 days/week = 14 kg/week</p> <p>Step sentence: A 42 kg bag of rice will last 42 kg / 2 kg per week = 21 weeks.</p> <p>Correctness of the step: Wrong</p> <p>Explanation: This step is wrong because it incorrectly calculates the duration the rice will last. Given the correct weekly consumption is 14 kg/week, the duration the 42 kg bag will last is:</p> $42 \text{ kg} \div 14 \text{ kg/week} = 3 \text{ weeks}$ <p>Correctness of the final answer: Wrong</p> <p>Explanation: The final answer of 21 weeks is incorrect. The correct answer is 3 weeks.</p> <p>Discrimination Verifier: </p> <p>Actor Model (refinement response):</p> <p>Let's break it down step by step:</p> <ol style="list-style-type: none"> Calculate the daily rice consumption: The household consumes 0.2 kg of rice per meal. Since there are 2 meals per day, the daily consumption is: 0.2 kg/meal \times 2 meals/day = 0.4 kg/day. Calculate the weekly rice consumption: The household consumes 0.4 kg of rice per day. Since there are 5 days per week, the weekly consumption is: 0.4 kg/day \times 5 days/week = 2 kg/week. Calculate how many weeks a 42 kg bag of rice will last: A 42 kg bag of rice will last: $42 \text{ kg} \div 2 \text{ kg/week} = 21 \text{ weeks.}$ <p>The answer is 21.</p> <p>Correctness Verifier: </p>

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

Figure 8: Example 1 of qualitative analysis. The actor's original response is incorrect. The model after SFT is unable to detect errors in the response, leading the actor's refinement response to retain the same errors. However, the model trained after Critique-RL identifies the errors in the original response and provides detailed, constructive suggestions for modification, leading to the correct refinement response.

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

<p>Problem: Dennis uses 1 pound of butter for every dozen croissants that he makes. He needs to make 6 dozen croissants. The grocery store currently has a promotion for buy one pound of butter get one half off. If the butter costs \$4.00 a pound, how much will it cost him to purchase 6 pounds of butter?</p>	
<p>Ground Truth Answer:</p> <ul style="list-style-type: none"> 1 pound of butter makes 1 dozen croissants and he needs to make 6 dozen so he needs $1 \times 6 = 6$ pounds of butter. The butter is \$4.00 a pound and currently on sale for buy one get one half off so every unit of 2 pounds of butter will cost $4 \times 1.5 = \\$6.00$. He needs 6 pounds of butter and the current promotion is for a unit of 2 pounds of butter so he needs to buy $6/2 = 3$ units worth of butter. The sale price is \$6.00 per 2 pounds of butter and he needs to buy 3 units of butter so $\\$6 \times 3 = \\18.00 worth of butter. <p>The answer is 18.</p>	<p>Actor Model (original response):</p> <p>Let's break it down step by step:</p> <ol style="list-style-type: none"> He needs to buy 3 pounds of butter because $6 / 2 = 3$. 3 pounds of butter would cost \$12 because $3 \times 4 = 12$. He will pay \$6 because $12 / 2 = 6$. <p>The answer is 6.</p>
<p>Critique Model after Critique-RL Stage I Training</p> <p>Critique Model (critique):</p> <p>Step sentence: He needs to buy 3 pounds of butter because $6 / 2 = 3$. Correctness of the step: Wrong</p> <p>Explanation: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. The calculation $6 / 2 = 3$ is incorrect. He needs to buy 6 pounds, not 3.</p> <p>Step sentence: 3 pounds of butter would cost \$12 because $3 \times 4 = 12$. Correctness of the step: Wrong</p> <p>Explanation: Although the calculation of the cost is correct (3 pounds at \$4 per pound equals \$12), the premise is wrong. Dennis needs 6 pounds of butter, not 3. The correct calculation should be 6 pounds at \$4 per pound, which equals \$24.</p> <p>Step sentence: He will pay \$6 because $12 / 2 = 6$. Correctness of the step: Wrong</p> <p>Explanation: The calculation $12 / 2 = 6$ is correct based on the promotion (buy one pound get one half off), but it does not apply to the correct amount of butter needed. Dennis needs 6 pounds, not 3. The correct calculation should be based on the full price of 6 pounds, which is \$24.</p> <p>Correctness of the final answer: Wrong</p> <p>Explanation: The final answer of 6 is incorrect. Based on the correct calculation, Dennis needs 6 pounds of butter at \$4 per pound, which totals \$24.</p> <p>Actor Model (refinement response):</p> <p>Discrimination Verifier: </p> <p>Let's break it down step by step: To determine how much it will cost Dennis to purchase 6 pounds of butter, we need to consider the following:</p> <ol style="list-style-type: none"> Butter Requirement: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. Promotion Details: The store has a promotion where you buy one pound of butter and get the next pound at half price. Cost Calculation: <ul style="list-style-type: none"> Full Price for 6 Pounds: Since the promotion applies to each pair of pounds, we need to calculate the cost based on the full price for 6 pounds. Cost for 6 Pounds: The cost of 6 pounds of butter at \$4 per pound is \$24. Therefore, the total cost for 6 pounds of butter is \$24. <p>The answer is 24.</p>	<p>Actor Model (original response):</p> <p>Let's break it down step by step:</p> <ol style="list-style-type: none"> He needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. The promotion does not change the fact that he needs 6 pounds. The calculation $6 / 2 = 3$ is irrelevant to the problem. 3 pounds of butter would cost \$12 because $3 \times 4 = 12$. Correctness of the step: Wrong Although the promotion is "buy one pound of butter, get one half off," Dennis needs 6 pounds of butter. The promotion applies to each pound he buys, so for every 2 pounds, he pays \$4 + \$2 = \$6. Since he needs 6 pounds, he can take advantage of the promotion 3 times (6 pounds / 2 pounds per promotion). Therefore, the total cost is $3 \times \\$6 = \\18. Step sentence: 3 pounds of butter would cost \$12 because $3 \times 4 = 12$. Correctness of the step: Wrong The calculation $12 / 2 = 6$ is incorrect. Based on the previous explanation, the total cost is $3 \times \\$6 = \\18. Therefore, he will pay \$18 instead of \$6. Correctness of the final answer: Wrong The final answer of 6 is incorrect. Given the promotion, Dennis will pay \$18 for 6 pounds of butter. Discrimination Verifier: <p>Actor Model (refinement response):</p> <p>Let's break it down step by step: To determine how much it will cost Dennis to purchase 6 pounds of butter, we need to consider the following:</p> <ol style="list-style-type: none"> Butter Requirement: Dennis needs to make 6 dozen croissants, and he uses 1 pound of butter per dozen. Therefore, he needs 6 pounds of butter. Promotion Details: The promotion is "buy one pound of butter, get one half off." This means for every 2 pounds of butter purchased, Dennis pays for 1 pound at full price and gets the second pound at half price. Cost Calculation: <ul style="list-style-type: none"> For every 2 pounds of butter, Dennis pays \$4 for the first pound and \$2 for the second pound, totaling \$6 for 2 pounds. Since Dennis needs 6 pounds, he can take advantage of the promotion 3 times (6 pounds / 2 pounds per promotion). Therefore, the total cost is $3 \times \\$6 = \\18. <p>The answer is 18.</p>

Figure 9: Example 2 of qualitative analysis. The actor's original response is incorrect. The model trained after Critique-RL Stage I is able to detect this error, demonstrating its discriminability. However, the model provides the actor with low-quality suggestion, causing the actor's refinement response to be incorrect. In contrast, for the same erroneous original response, model trained after Critique-RL Stage II not only detects the error but also offers a constructive suggestion, ultimately leading to the correct refinement response, demonstrating the advantage of two-stage RL process.

1233

1234

1235

1236

1237

1238

1239

1240

1241