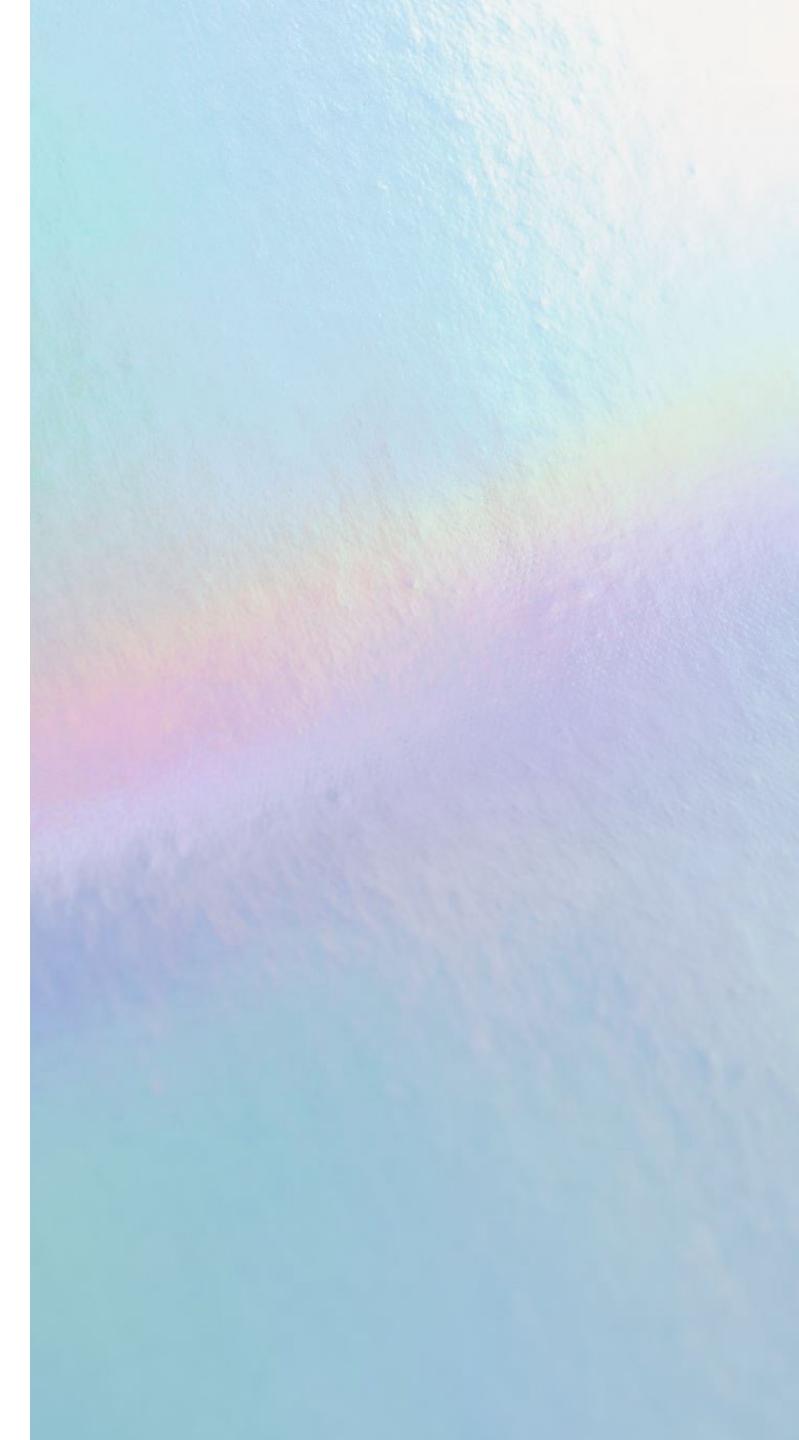


---

# **ENGLISH TEXT COMPLETION PROBLEM ANSWERING USING GPT-2**

Shangyu Chen, Yuqi Yan

CSE 561A: Large Language Models



---

# Introduction

- Background:
  - The High School Entrance Exam and College Entrance Exam
  - Grammar, vocabulary, and comprehension.
  - Automatically generating answers, enhancing accuracy and efficiency.

# Introduction

- Model: GPT-2 (Text Generation Model), GPT-3.5 (dataset generating)
- Main works:
  - Exploring datasets for fine-tuning
  - Problem converting to problem data structure using LLM
  - Simple answering test

# Problem Dataset

- Source
  - Real problems from high school/college entrance exam
- Text example

30. The student was just about to \_\_\_\_\_ the question, when suddenly he found the answer.

A.arrive at

B.work out

C.Stick to

D.give up

答案 : D

# Problem Dataset

- Data cleaning

- Problem format is not uniform, there are some errors

77. I need  food. Would you like to give me  apples

- A. more some; more two
- B. some more; two more
- C. more some; two more
- D. some more; more two

185. When speaking in public, \_\_\_\_\_.

- A. speech should be understood
- B. speech should be easy to understand
- C. one should make himself understood
- D. one should make listeners understand

151.-Let's go to the concert this evening. It's  be excellent.

-I'd like to, but I have something to do.

- \_\_\_\_! It is the last time. We'll have no chance to enjoy it.

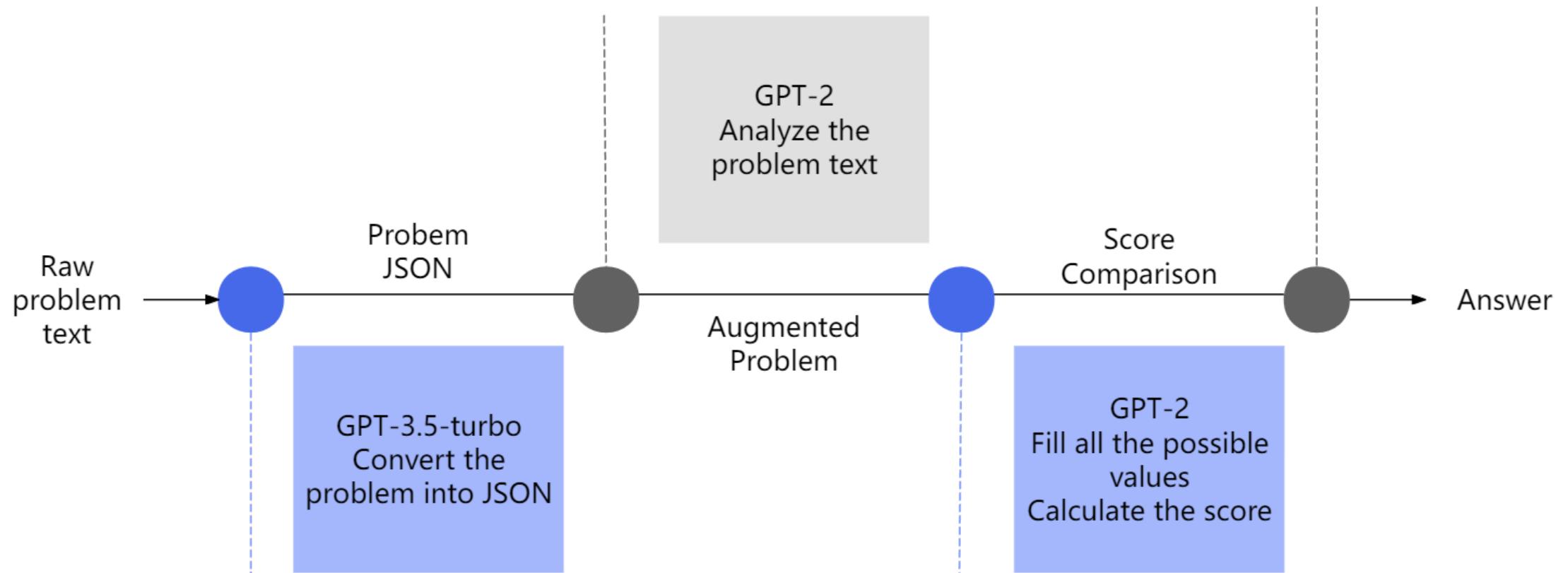
- A. What a pity
- B. Not at all
- C. All the best
- D. Never mind

# Data structure

- Problem JSON

```
{  
    "question": "After watching {} TV, she played {} violin for an  
hour.",  
    "options": [  
        ["", ""],  
        ["the", "the"],  
        ["the", ""],  
        [ "", "the"]  
,  
    "answer": 3  
}
```

# Pipeline



# Data processing: Text to JSON

## Few-shot processing

- GPT for coding: GitHub Copilot
  - **Prompt:** Convert the problem text to JSON.
  - // All the blanks will be substituted to "{}" in the question.
  - // The answer part, 0 is "A", 1 is "B", 2 is "C", 3 is "D", representing the option index.
  - Example of the question: ...

The image shows a code editor with two tabs open, both titled "4-2.txt".

**Left Tab (4-2.txt):**

```
1  ``json
  2      [
  3          {"question": "I invited Tom and Ann to dinner, but {} of them came.", "options": ["neither", "both", "either", "none"], "answer": 0},
  4          {"question": "Does {} matter if he can't finish the job on time?", "options": ["this", "that", "he", "it"], "answer": 3},
  5          {"question": "The experiment was {} easier than we had expected.", "options": ["more", "much more", "much", "more much"], "answer": 2},
  6          {"question": "The volleyball match will be put off if it {} .", "options": ["will rain", "rains", "rained", "is raining"], "answer": 1},
  7          {"question": "The library needs {}, but it'll have to wait until Sunday.", "options": ["cleaning", "be cleaned", "clean", "being cleaned"], "answer": 0},
  8          {"question": "Mary {} a dress when she cut her finger.", "options": ["made", "is making", "was making", "makes"], "answer": 2},
  9          {"question": "I learned to {} a bicycle as a "}
  10         ]
  11
  12     // // All the blanks will be substituted to "{}" in the question.
  13     // The answer part, 0 is "A", 1 is "B", 2 is "C", 3 is "D" representing the option index.
```

**Right Tab (4-2.txt):**

```
200 35. Does ___ mat
  201 job on time?
  202 A. this B. that C.
  203 Answer: D
  204
  205 36. The experiment was
  206 expected.
  207 A. more B. much more
  208 Answer: C
  209
  210 37. The volleyball match w
  211 A. will rain B. rains
  212 raining
  213 Answer: B
  214
  215 38. The library needs ___
  216 until Sunday.
  217 A. cleaning B. be cleaned
  218 cleaned
  219 Answer: A
  220
  221 39. Mary ___ a d
  222 finger.
  223 A. made B. is making
  224 Answer: C
  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
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  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
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  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
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  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
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1747
  1748
  1749
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1760
  1761
  1762
  1763
  1764
  1765
  1766
  1767
  1768
  1769
  1770
  1771
  1772
  1773
  1774
  1775
  1776
  1777
  1778
  1779
  1780
  1781
  1782
  1783
  1784
```

# Data processing: Text to JSON

- Different types of questions are provided with different examples
  - For questions with only one blank: Convert the blank into {}
  - For questions with multiple blanks: Convert separators into arrays
  - For questions with errors, perform corrections
- Initially, generate and correct simultaneously to train on various possibilities
- In the later stages, generation can be automated, but it is important to pay attention to the distance between the position of the question text and the text in the generation area.

# Results: Data Processing

- A total of 1579 problems

	<b>Numbers of Problems</b>
JSON format error	10
Inconsistent with the original text	7
Correctly converted	1562

# GPT-2 finetuning

- Raw model testing
  - Calculating the loss, selecting the one with the least loss.

Model	Valid problems	Correct	Correct rate
GPT-2	1562	1077	69.0%
GPT-2-medium		1164	74.5%
GPT-2-large		1175	75.2%
GPT-2-xl		1189	76.1%

# GPT-2 finetuning

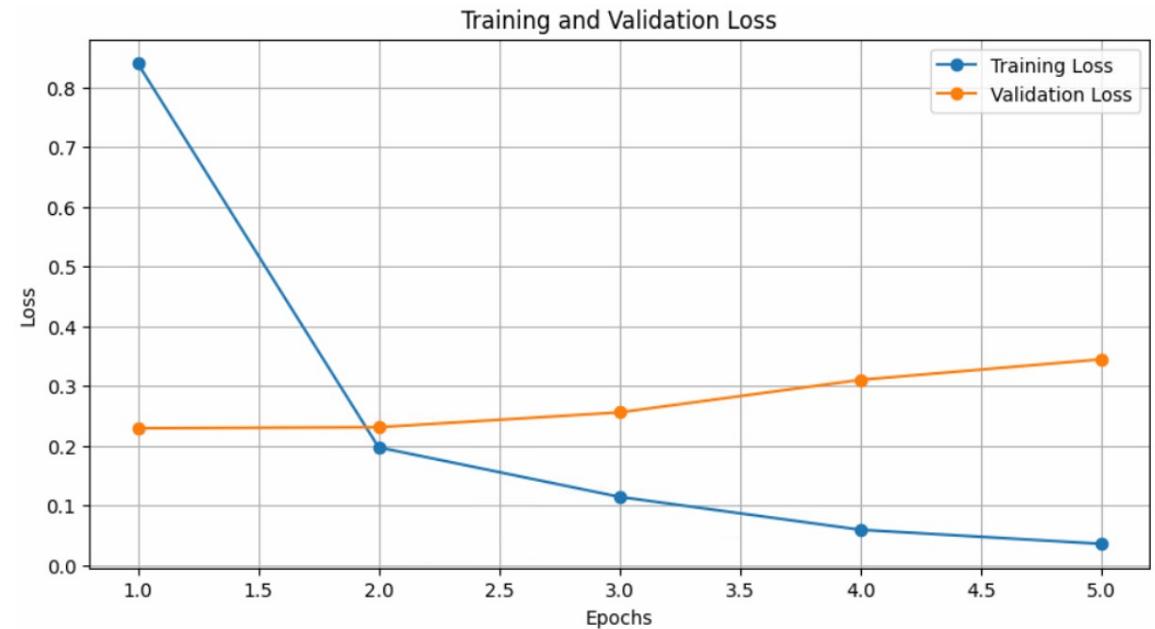
- Train-test splitting:
  - 4:1:1 <-> train, validation, test

```
[5]: df_train = pd.read_csv("./data/train_sentence.csv")
      df_train.head(5)
```

	index	sentence
0	0	There are many other ways to advertise besides...
1	1	Staring into his eyes, I could see a puzzled l...
2	2	His story proved true.
3	3	The petrol tank is empty. I should have filled...
4	4	The scientist buried himself in his studies in...

# GPT-2 finetuning

- Unexpected results:
  - Training loss get closer to 0
  - Validation loss keeps going up



Model	Test problems	Correct	Correct rate
Raw GPT-2	391	273	69.82%
Finetuned		258	65.98%

# GPT-2 finetuning

- Overfitting
  - Use problem text directly. Text is too short.
  - Remembering the text instead of analyzing the text.
- Analyzing the problem that GPT-2 can and cannot solve

# Data augmenting

- Creating better dataset: Problems that GPT-2 can't solve
- Logistic problems
  - The sentence is: It would {} us a {} time, but fortunately, we didn't.
  - The best option is: ['have taken', 'long']
  - ['take', 'along']: -3.885208
  - ['take', 'longer']: -3.258977
  - ['have taken', 'longer']: -2.959896
  - ['have taken', 'long']: -2.545516

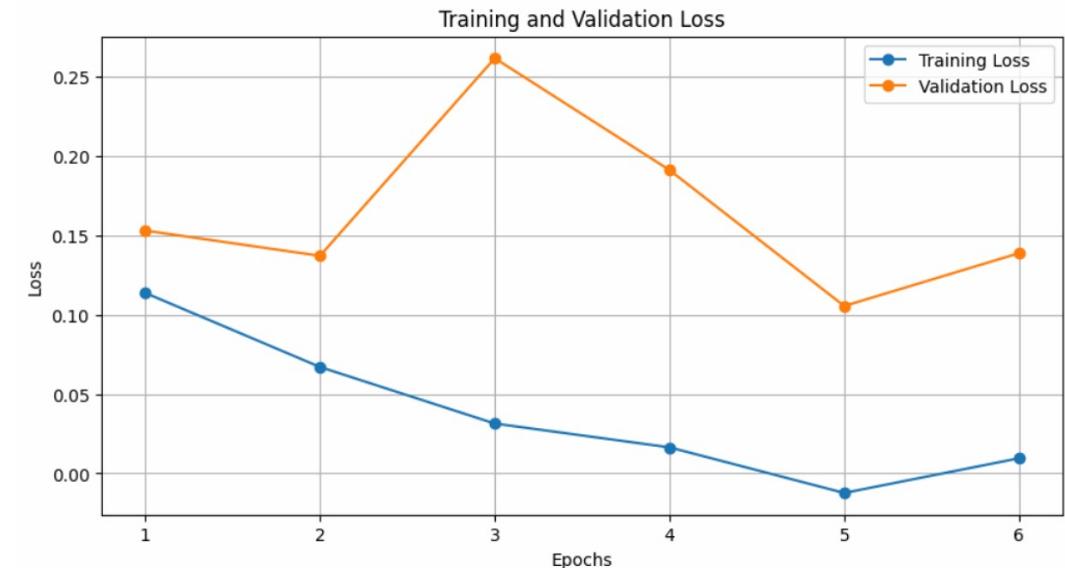
The syntax is correct, but it's not logical.

# Data augmenting

- Augmenting the problem text with the same logic (GPT-3.5, translating)
- Prompt: Generate some similar logical text for my dataset (at least 10)
- Waste paper can be reused for making new paper, which is called recycling. Used paper can be processed to create new paper products, a process known as recycling. Recycling involves reusing waste paper to manufacture fresh paper, reducing the need for new raw materials...

# Data augmenting

- Hyperparameters:
  - 0.15 dropout rate, 1e-6 learning rate
- Better results:
  - Training loss get closer to 0
  - Better validation loss



Model	Test problems	Correct	Correct rate
Raw GPT-2	391	273	69.82%
Finetuned w/ raw data		258	65.98%
Finetuned w/ augmented data		289	73.91%

# Using chain-of-thought

---

- Dataset included the relationship of the logic.
- CoT dataset creation:
  - using GPT-3.5-turbo



You

Prompt: Please explain the logistics of the problem, the comparison of the options, and infer the answer. The answer is 'A. envies'

---Mary looks unhappy.  
---Yes, she always \_\_\_ our success.  
A. envies  
B. cheers  
C. admires  
D. stops

< 2 / 2 > ⏪



ChatGPT

The logistics of the problem involve understanding the context of the conversation, where one person suggests taking a coffee break and the other responds in a way that reflects agreement or acknowledgment of a previously expressed sentiment.

Comparing the options, "You said it" best fits the situation, indicating agreement with the suggestion of taking a break after working for hours.

Therefore, the answer is D. You said it.



ChatGPT

The logistics of the problem involve recognizing Mary's unhappiness and understanding the reason behind it, which is likely related to her attitude towards the success of others.

Comparing the options, "envies" stands out as the most appropriate choice since it reflects Mary's negative feelings towards the success of others.

Therefore, the answer is A. envies.

# Using chain-of-thought

- Finetuning Dataset Composed:
  - Question:
    - -Mary looks unhappy. -Yes, she always <|missing|> our success.
  - Answer:
    - The logistics of the problem involve recognizing Mary's unhappiness... Comparing the options, “**envies**” stands out as the most appropriate choice since it reflects Mary's negative feelings towards the success of others. The answer is “-Mary looks unhappy. -Yes, she always **envies** our success. ”

# Using chain-of-thought

- After input the question, 100 words are generated to complete the sentence logic
- Manually add the text after generation, “The answer is”, with the complete sentence, and calculate the probability.

A bit better, but not better than the medium model.

# Results

Model	Test problems	Correct	Correct rate
Raw GPT-2	391	273	69.82%
Finetuned w/ raw data		258	65.98%
Finetuned w/ augmented data		289	73.91%
Adding chain of thought		292	74.68%
GPT-2 medium		301	76.98%

# Conclusion

---

- Data sets that are too simple are of low quality and can easily lead to overfitting.
- Data sets containing **logical analysis** can be helpful for problem solution – CoT analyzed implicit logics
- GPT-2 has limited generation capabilities, more complex models are supposed be considered in the future work.

---

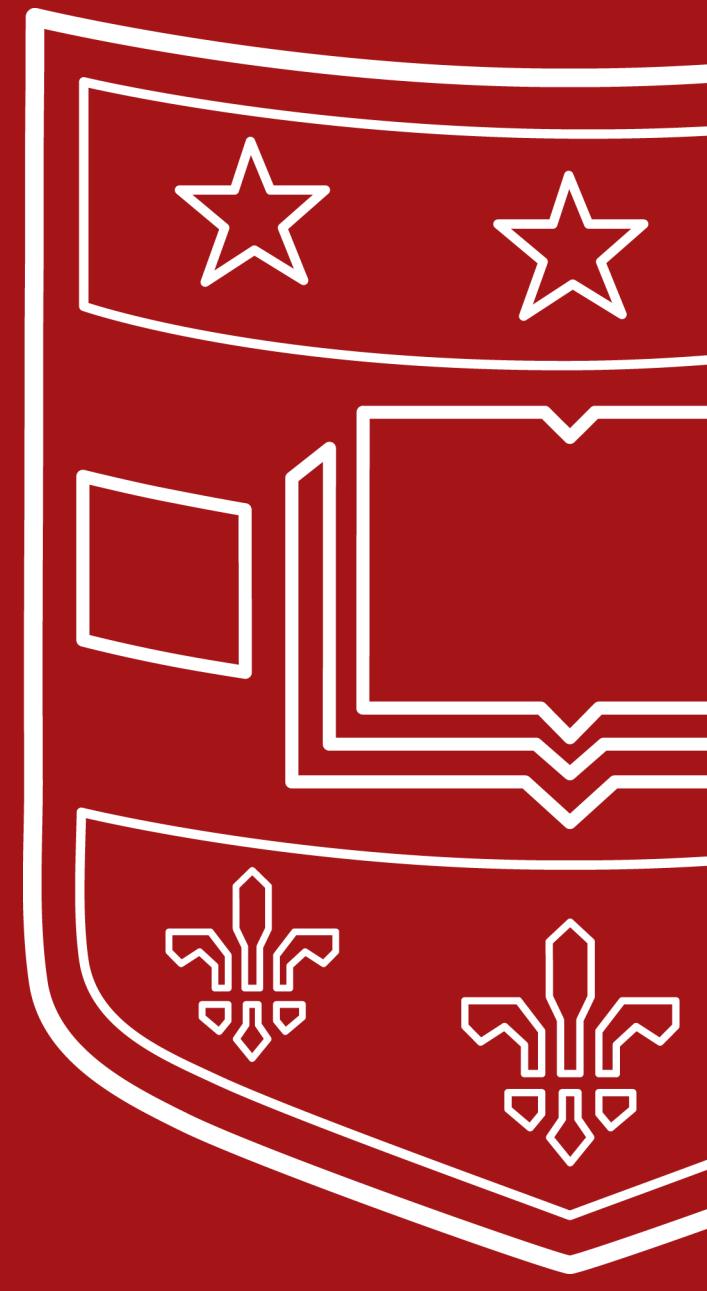
**Q&A**

**Thanks!**

# Enabling Large Language Models to Solve Divide and Conquer Problems with Recursion

Kyle Montgomery  
Kyle Stein

 Washington University in St. Louis

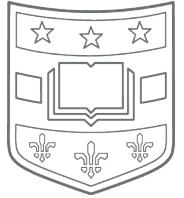




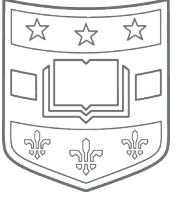
# Introduction

- **Problem:** Length Generalization
  - LLMs struggle to generalize to problems more difficult than those seen training
- We focus on algorithmic tasks, namely divide-and-conquer problems (mergesort, exponentiation, etc.)
  - Breaking down problems into smaller sub-problems
  - Solving each sub-problem
  - Combining sub-problem solutions to solve the original problem
- **Key Idea:** We exploit the recursive nature of these problems in order to teach LLMs to solve these problems recursively.

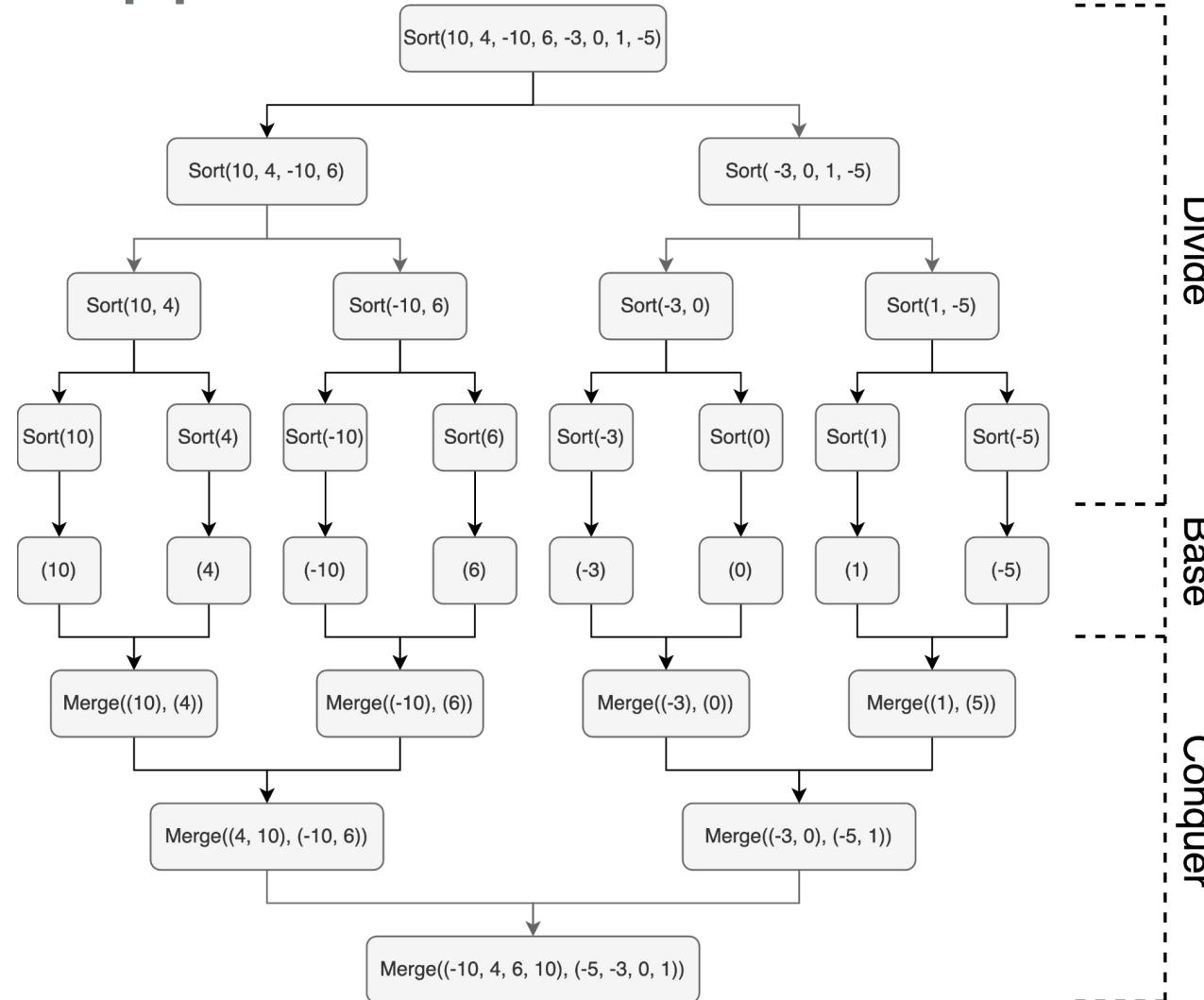
# Related Work



- Several works have studied the ability of LLMs to length-generalize on algorithmic tasks (Anil et. al., 2022; Dziri et. al., 2023; Zhou et. al., 2023; Pasewark et. al., 2024). They all note that LLMs struggle to solve instances of a task that are more complex than seen during training.
- Length generalization is an important problem: if LLMs fail to generalize to tasks more complex than those seen during training, it could be a huge roadblock on the road towards super-intelligence.
- This work is also inspired by recent literature on LLM tool usage (Schick et al., 2023; Qin et al., 2023; Paranjape et al., 2023), only in our case the LLM is using itself as the tool!



# Overview of Approach



Recursively call and solve each sub-problem in it's own context!



# Recursive Approach - Training

Our recursive training data includes 3 components:

- Divide steps:
  - input: “sort(10, 4, -10, 6, -3, 0, 1, -5) = ”
  - output: “merge(sort(10, 4, -10, 6), sort(-3, 0, 1, 5))”
- Base-case steps:
  - input: “sort(10) = ”
  - output: “10”
- Conquer steps:
  - input: “merge((-10, 4, 6, 10), (-3, 0, 1, 5)) = ”
  - output: “(-10, -3, 0, 1, 4, 6, 5, 10)”

**Claim:** This training data alone is sufficient to teach a model mergesort.



# Recursive Approach - Inference

```
def recursive_generate(model, tokenizer, prompt)
    output = generate(model, tokenizer, prompt) # divide
    if not call in output: # base case
        return output
    for call in extract_calls(output): # loop through calls (e.g., sort calls)
        result = recursive_generate(model, tokenizer, call)
        output.replace(call, result)
    output = generate(model, tokenizer, output) # conquer
    return output
```

Pseudocode for recursive inference.



# Experiment Setup

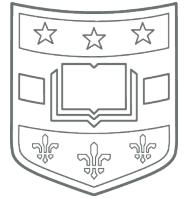
We're considering two divide-and-conquer problems:

- MergeSort (e.g.,  $\text{sort}(10, 4, -10, 6, -3, 0, 1, -5) = ?$ )
- Exponentiation (e.g.,  $9^{14} = ?$ )

For each problem, we fine-tune LLaMA-7B using low-rank adapters (LoRA).

We train on problems up to a certain complexity, evaluate on more complex problems to assess generalization performance.

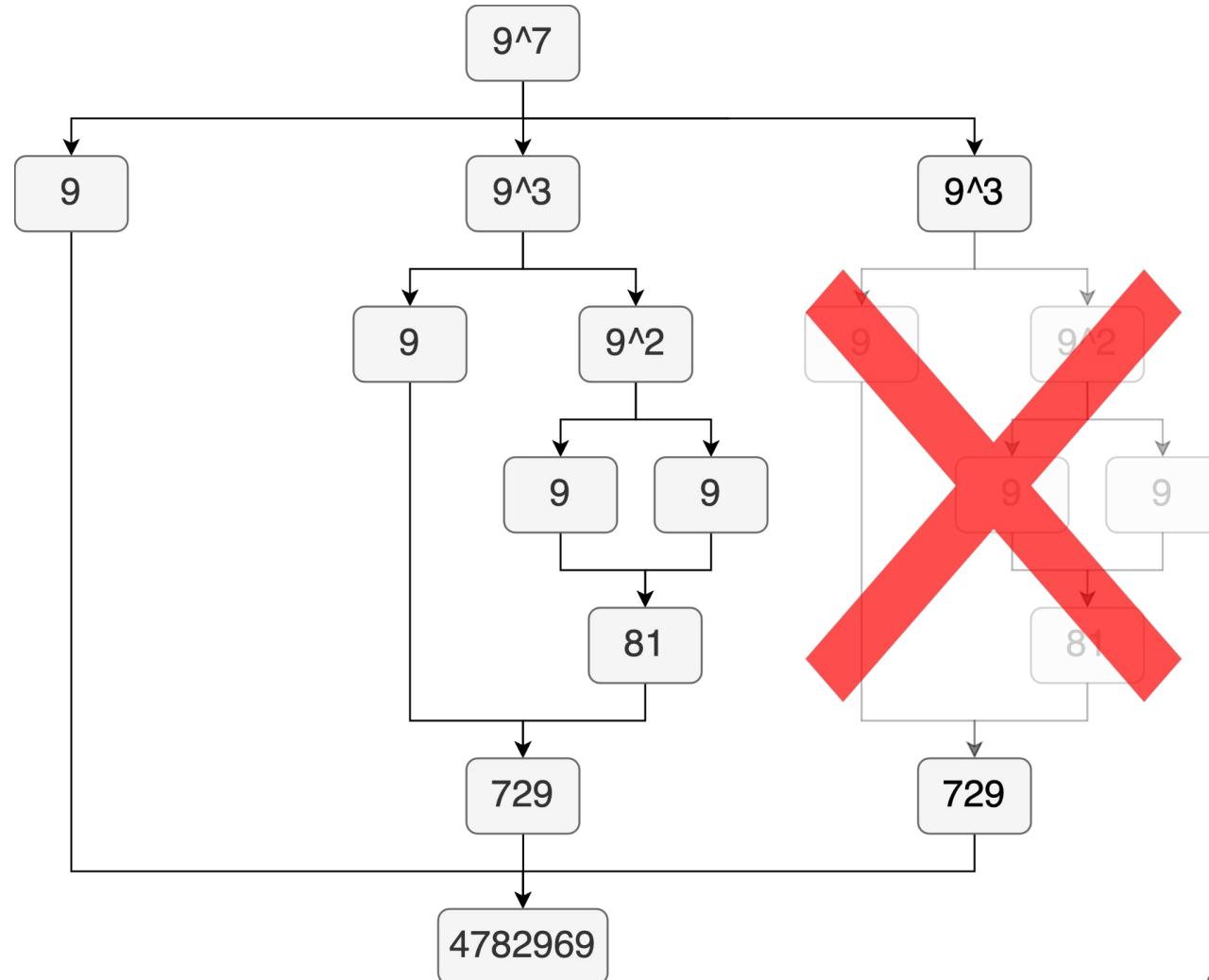
- e.g., on Mergesort, the training data includes arrays upto length 15, while the testing data includes arrays upto length 30!



# Experimental Setup – Optimizations

In practice, we found the following two optimization helpful:

- Limiting the depth of the recursion stack.
  - The depth of the recursive stack should never exceed  $\text{ceil}(\log(\text{exponent})) + 1$ .
- Caching solutions to sub-problems.
  - Once we compute  $9^3$ , we'll never need to compute it again!
  - Generation time decreases with scale!





# Comparison Methods

**Baseline:** immediately prompt for the solution with no intermediate reasoning.  
steps (e.g.,  $\text{sort}(10, 4, -10, 6, -3, 0, 1, -5) = (-10, -5, -3, 0, 1, 4, 6, 10)$ )

**Scratchpad:** permit intermediate reasoning steps, but only in the same context.

[start] Splitting: [10, 4, -10, 6, -3, 0, 1, -5]

  [left] Splitting: [10, 4, -10, 6]

    [left] Splitting: [10, 4]

      [left] Base case: [10]

      [right] Base case: [4]

    [left] Merging: [4, 10]

    [right] Splitting: [-10, 6]

      [left] Base case: [-10]

      [right] Base case: [6]

    [right] Merging: [-10, 6]

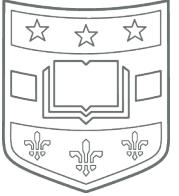
  [left] Merging: [-10, 4, 6, 10]

  [right] Splitting: [-3, 0, 1, -5]

  ...

  [right] Merging: [-5, -3, 0, 1]

[start] Merging: [-10, -5, -3, 0, 1, 4, 6, 10]

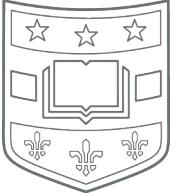


# Results – Mergesort

We trained on randomly generated arrays up to length 15.

Array Length	1	5	10	15	20	25	30
Baseline	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Scratchpad	1.00	1.00	1.00	1.00	0.72	0.06	0.00
Recursive	1.00	1.00	1.00	1.00	0.98	0.62	0.22

The fine-tuned baseline turned out to be stronger than expected. Zhou et. al., 2023 also explored the sorting problem as well, and found the transformer architecture is well-equipped for sorting.



# Results – Exponentiation

We trained on randomly generated exponents up to 8.

Exponent	0	3	6	9	12	15	18	21
Baseline	1.00	1.00	1.00	0.08	0.12	0.14	0.00	0.00
Scratchpad	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00
Recursive	1.00	1.00	1.00	0.16	0.34	0.00	0.16	0.00

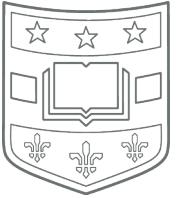
Again, the baseline is strong, but the recursive method is able to outperform both the baseline and scratchpad approaches on exponents of 9, 12, and 18.

Interestingly, the scratchpad and recursive methods struggle to generalize on problems with an odd exponent, while this does not appear to be the case for the baseline method.



# Discussion of Results

- Baseline performance on sorting suggests transformers models have an inherent model ability to solve sorting problems without explicit reasoning.
- The recursive method generalizes significantly better than the scratchpad approach, highlighting the strength of solving subproblems in new contexts.
- Limitations of recursive approach:
  - We noticed that the majority of the errors made are during the conquer steps (e.g., merging two sorted arrays, multi-digit multiplication).
  - We hypothesize that performance would improve if we trained a separate model for these conquer steps, or leveraged additional recursive techniques (e.g., Karatsuba's fast multiplication algorithm)



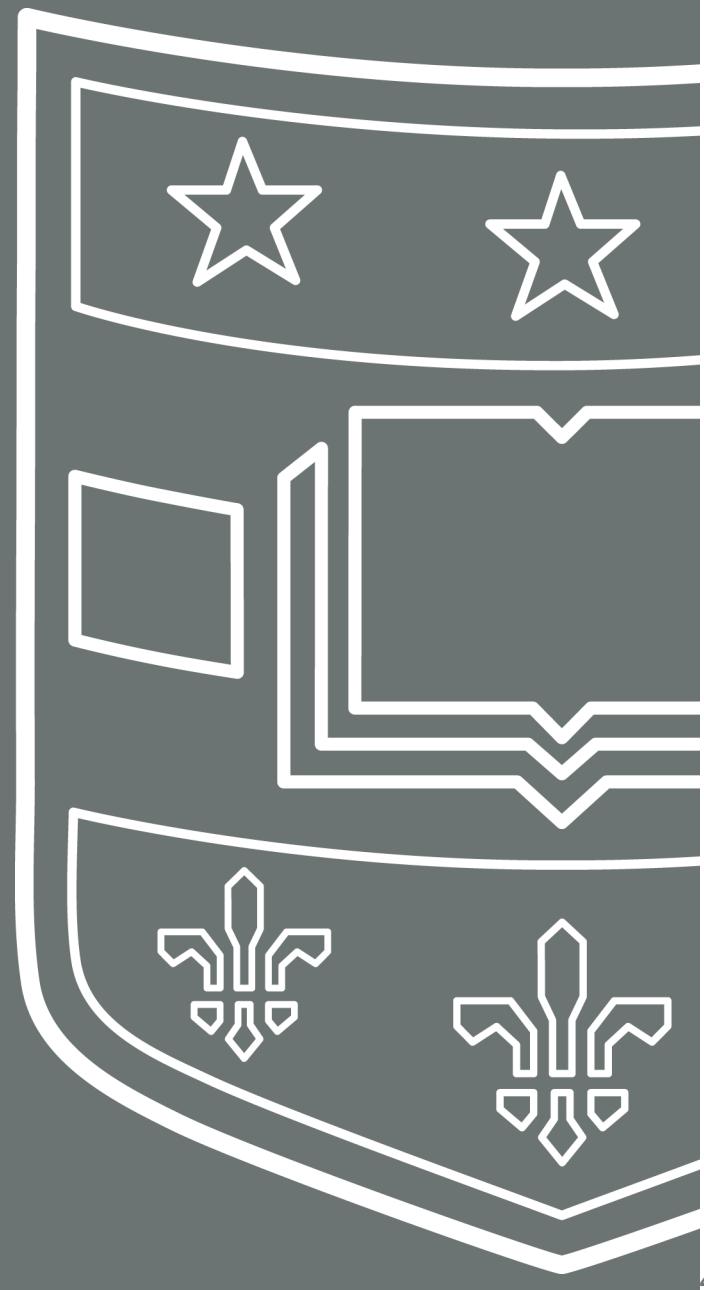
# Conclusion

- Aimed to explore the ability of LLMs to length generalize on divide-and-conquer problems with the help of recursion.
- Constructed a novel training and inference pipeline to enable LLMs to solve algorithmic divide-and-conquer problems recursively.
- Compared our method to baseline and scratchpad approaches on two problems (MergeSort and Exponentiation).
- Results are mixed, but suggest that future research is needed to fully understand the recursive abilities of LLMs.

# Thank You!



Washington University in St. Louis



# 561 Demo

Zhexiao and Jason

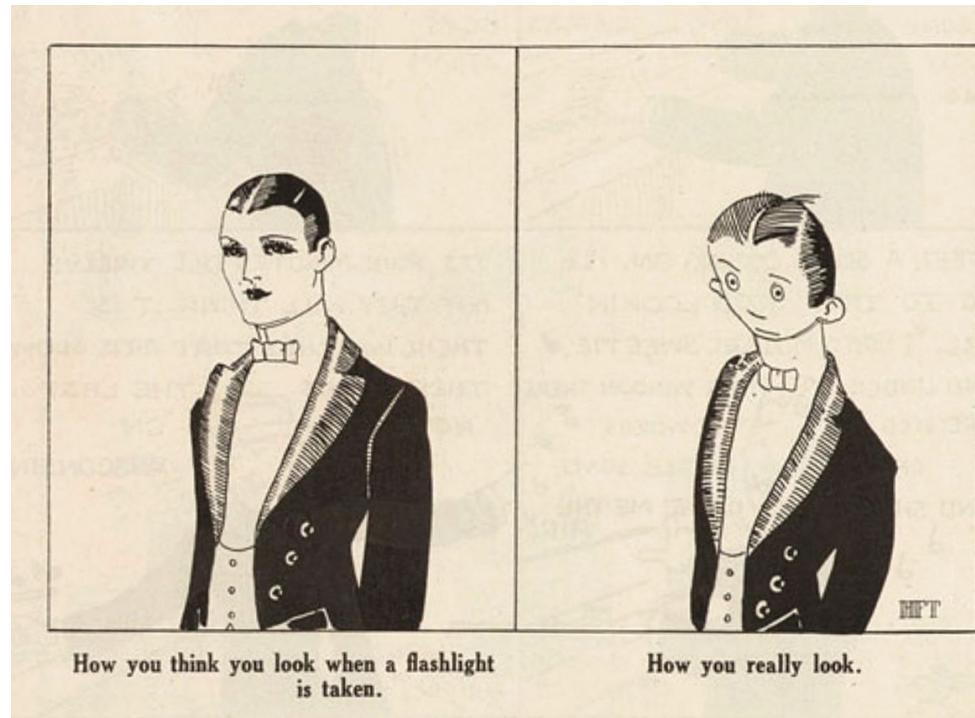
# Background Information

- Memes are behaviors with **symbolic** meaning.
- They can be expressed in many ways, from physical action, text, to **images**.
- Bigger screens and faster internet make meme images **practical**.
- Like emojis, meme images does **not** require typing!



# Motivation: choosing memes can be tricky

- Trendy and popular memes **change** quickly and often
- 🔎 The **message** of a meme image can be layered and complex
- The key is to match **intent** ❤️ with **context** 🌎
- ⚠️ The **same** meme image seem clever 😎, funny 😂, or rude 🤬
- ...like 'culture' (the word)



# Problem Setting: Image Customization & Retrieval

- We **decompose** a meme image into the image  itself and its accompanying caption 
- We are given a **library** of 300 memes
-  The user provides a **caption**
- The AI agent retrieves the best  $k$  meme images

Meme Top Text

Meme bottom text

# Example output

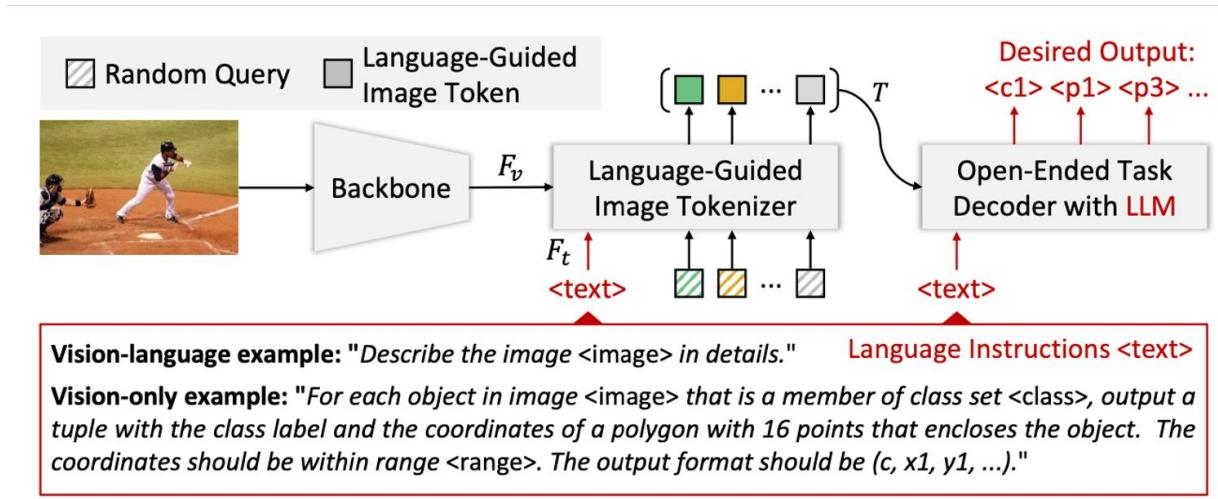
Caption	Model output (k = 3)	Not in top k
Very pi <sep> much digits	  	

🤔 There's **no perfect** meme image for a caption

😬 But some meme images and caption certainly **match poorly**

⚖️ It's more useful if the user can **choose** from a good selection of meme images, based on their **preference** and **need**

# Related Work: Vision Language Models



- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. arXiv preprint arXiv:2204.14198, 2022. 2, 4
- [2] Wenhui Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, et al. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks. arXiv preprint arXivL2204.14198v2 2023
- [3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. CoRR, abs/2304.08485, 2023.
- [4] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-Any Multimodal LLM. CoRR abs/2309.05519. 2023

# Related Work: Text-to-Image Customization



# Related Work: Few-Shot learning

- Few-shot learning studies how ML models perform well on unseen tasks with a few examples
- Optimization-based methods like MAML [1] are versatile but have high memory footprint. Later work like iMAML [2] alleviates this.
- Nonparametric methods like the Siamese networks [3] don't require differentiating over the model's optimization path, but are less well-rounded (e.g. no regression)

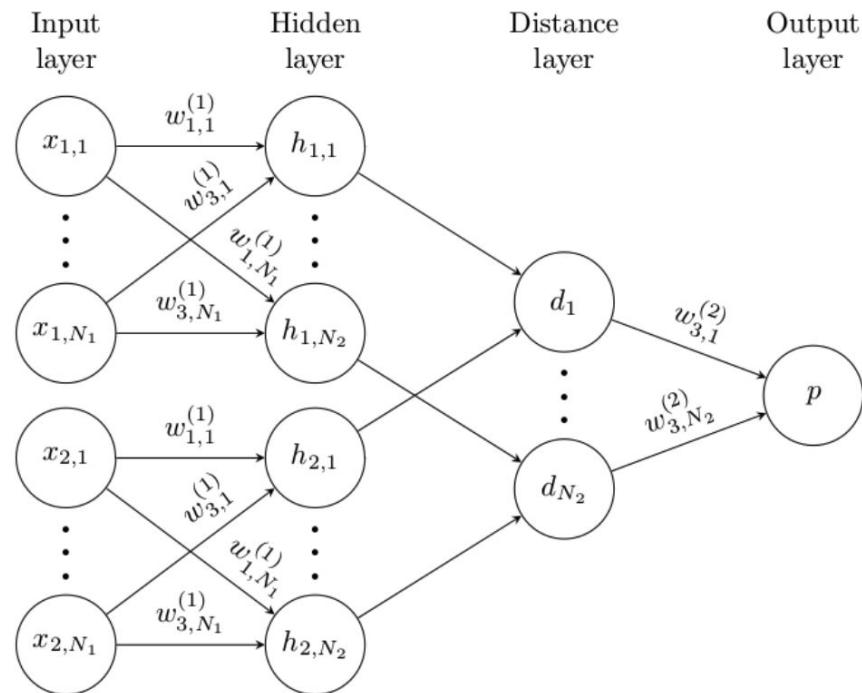
[1] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. International Conference on Machine Learning (ICML), 2017.

[2] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, Sergey Levine. Meta-Learning with Implicit Gradients. 2019. Conference on Neural Information Processing Systems.

[3] Koch, G., Zemel, R., and Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In ICML deep learning workshop, volume 2. Lille, 2015.

# Related Work: Siamese Architecture

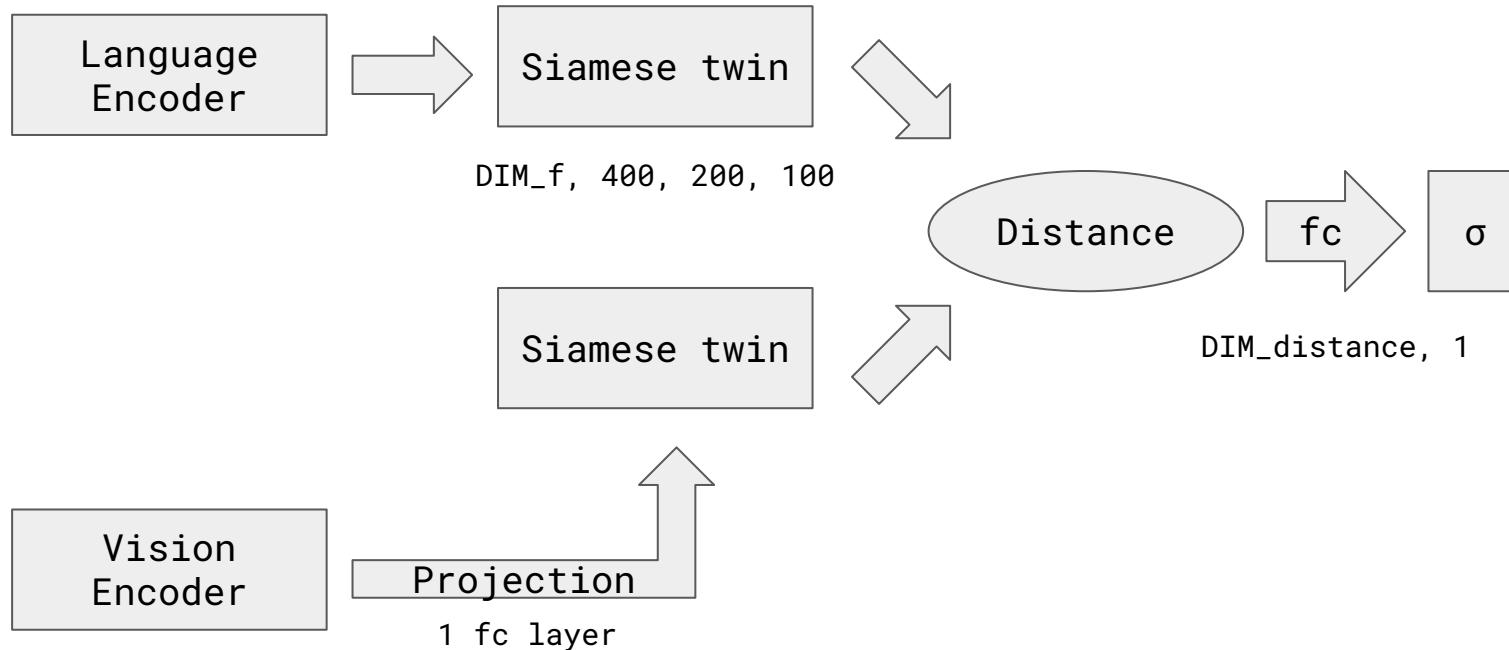
- One network functions as two networks with shared weights
- Nonparametric
- Common usage: human face recognition, signature verification [1]



[1] Bromley, Jane, Bentz, James W, Bottou, Léon, Guyon, Isabelle, LeCun, Yann, Moore, Cliff, Sauckinger, Eduard, and Shah, Roopak. Signature verification using a siamese time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7 (04):669–688, 1993.

# Model Architecture

Siamese network with additional projection layer



# Method Design

- ❖ 7:1:1 split for training, validation, and testing
- ❖ Training: binary classification on +positive and -negative (meme image, caption) pairs. The model's output is **one number** in  $[0, 1]$
- ❖ Inference: given a caption, select **k** images with the highest **similarity score** (i.e. the model's output)
- ❖ Optimization method: AdamW
- ❖ Different vision and language **encoders**
- ❖ Two **distance** operations: concatenate (FaceNet) vs weighted L1 (Koch et al)
- ❖ Additional **One-shot** experiment

# Experiment Results: encoders & distance

Best encoder combination: Inception V3 & Sentence Transformers

Accuracy (percentage )	Sentence Transformer s (dim = 384)	CLIP (dim = 512)	BERT (dim = 768)
Resnet50 (dim = 2048)	<b>87, 86</b>	69, 71	85, 82
VGG 16 (dim = 2048)	80, 81	73	78, 77
Inception V3 (dim = 512)	<b>88, 89</b>	86, 83	86, 88

Format:  
(concat, weighted L1)

Baselines:

kNN: 60

Gemini: 95

# Experiment Results: one-shot test

7:1:1 split of train/validate/test tasks

Ours: 81

kNN: 50

Gemini: **95**

Prompt setup for Gemini:

"[caption]"

*Is the above a good match for the following image?*

*Answer only yes or no*

[meme image]

👉 Gemini gives yes-no answer but follows up with explanation. We do not consider the explanation when computing accuracy

# Result Analysis

1. BERT-based models outperform CLIP (superior general textual understanding) [1]
2. Sentence Transformers (Sentence BERT, which coincidentally also used Siamese architecture) generates **sentence-level** embeddings, whereas BERT focus more on **individual words** and their context
3. Inception V3 is the best due to supporting multi-scale feature extraction
  - a. A meme image's **salient features** can exist at different **scales & positions**
4. Small network size, relative to the encoders (and most VLLMs), limits performance (both standard and few-shot)
  - a. Distance calculation did not make a consistent (or significant) difference
  - b. Significantly underperforms a large multimodal LM (Gemini)

[1] Chen, Zhihong; Chen, Guiming Hardy; Diao, Shizhe; Wan, Xiang; Wang, Benyou. On the Difference of BERT-style and CLIP-style Text Encoders. ACL, 2023.

# Enhancing Information Retrieval through Large Language Models

Zhengyang Xiao, Wenyu Li



STUDENT AFFAIRS





# Motivation & Intro

- The challenge of information retrieval varies across different domains due to the diversity of knowledge. This research focuses specifically on synthetic biology.
- Synthetic biology has gained significant attention for its ability to produce biological products through genetic engineering. Research in this field is increasing but remains fragmented and difficult to integrate.
- LLMs could potentially assist in synthesizing information from various research papers in synthetic biology, supporting holistic understanding and aiding research decisions.
- Information retrieval from scientific literature involves complex tasks beyond simple extraction, challenging even for biology PhD students in terms of comprehension and organization.
- From an LLM's perspective, the task divides into understanding prompts and identifying relevant texts, then inferring information through context and multiple sentences.

# Background



- Large language models (LLMs) have advanced text processing capabilities, enabling tasks like following user prompts and generating corresponding outputs.
- LLMs can achieve zero-shot learning, showing potential signs of general artificial intelligence, which is useful for extracting valuable information from texts.
- After receiving a specific prompt, LLMs can independently decide and efficiently retrieve information that interests users.



# Related works

## Application

- Legal case retrieval
- Biomedical records
- Visual question and answer
- General scientific knowledge
- “Foundation model”

## Information Retrieval techniques

- Dual encoder
- Retrieval-augmented generation
- Knowledge distillation
- Adapter layers
- Structured knowledge base

## Current challenges

- Lack of domain-specific knowledge
- Knowledge boundaries (multi-disciplinary)
- Knowledge conflicts
- Machine Hallucination

# Method



Compare LLM RAG capabilities

Obtained 7 synthetic biology  
research articles.

LLM try to extract experiment data  
and organize into data tables.

These models are compared: GPT-4,  
Bard, ChatGLM, Qwen (1 and 1.5  
with different parameter sizes)

Supervised Fine tune with QLoRA

Construct a dataset of 200 data  
points

Train existing model on huggingface  
Load new model for further testing of

1 prompt

Compare results



# Dataset construction & details

Prompt template:

```
<s>[INST] <<SYS>>  
{ system_prompt }  
<</SYS>>
```

```
{ user_message } [/INST]
```

# LoRA attention dimension = 64

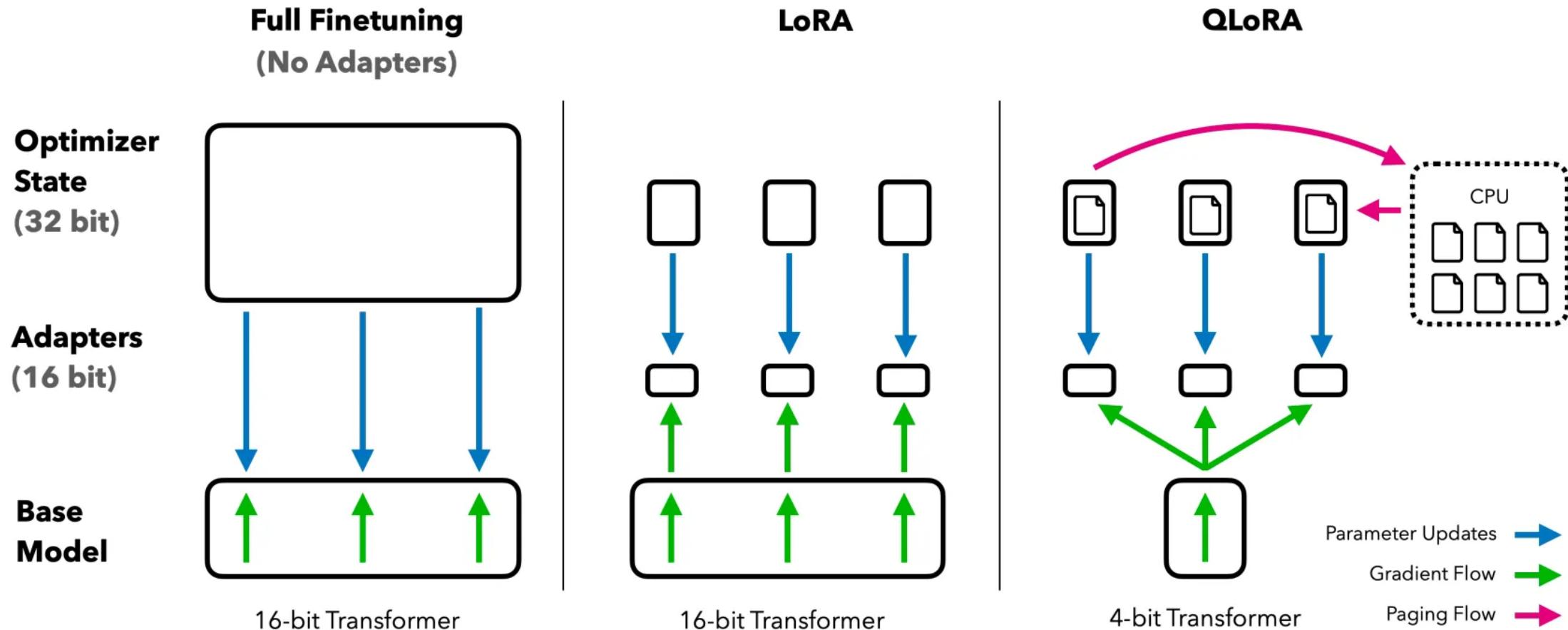
# Alpha parameter for LoRA scaling = 6

# Dropout probability for LoRA layers = 0.1

target modules: "q\_proj", "k\_proj", "v\_proj", "o\_proj", "gate\_proj"

GPU used: V100; total runtime: 2h/run

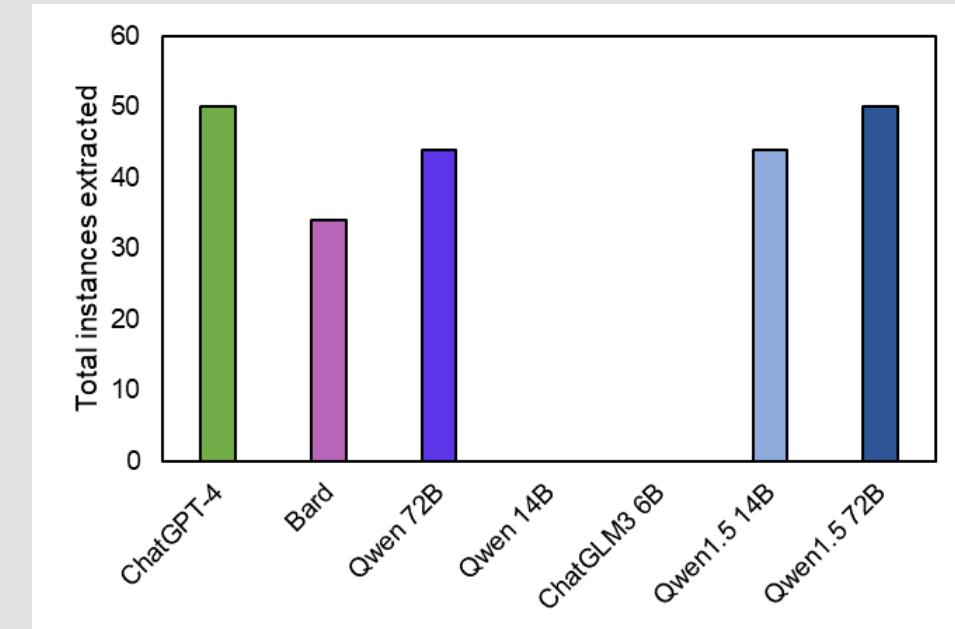
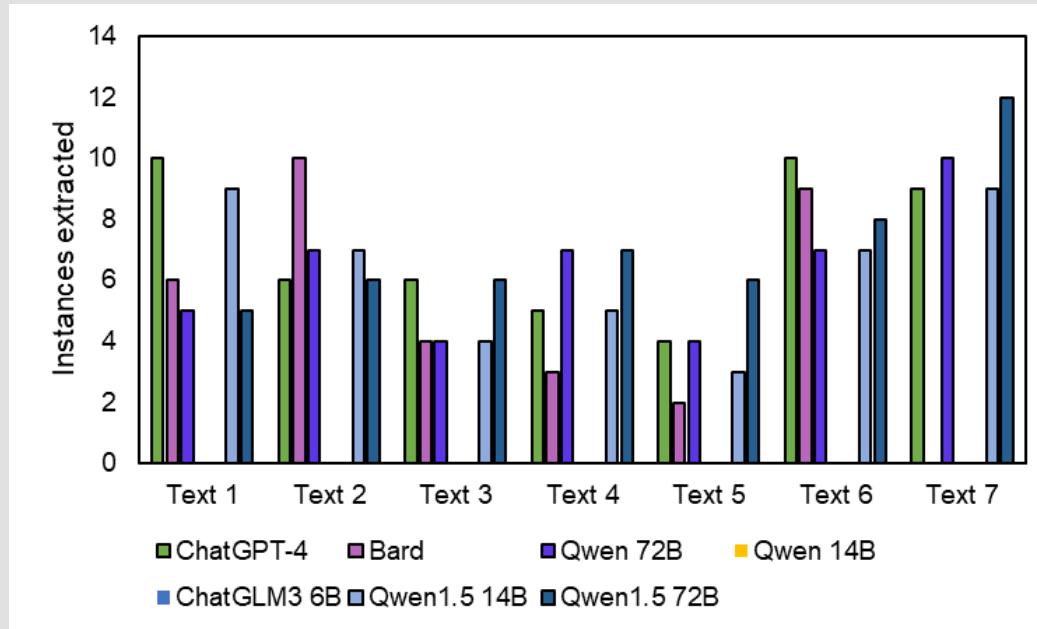
# QLoRA: Efficient Finetuning of Quantized LLMs, Dettmers et al.



**Figure 1:** Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

# Result: compare LLM information retrieval task

Data extracted in each text, best of three trials

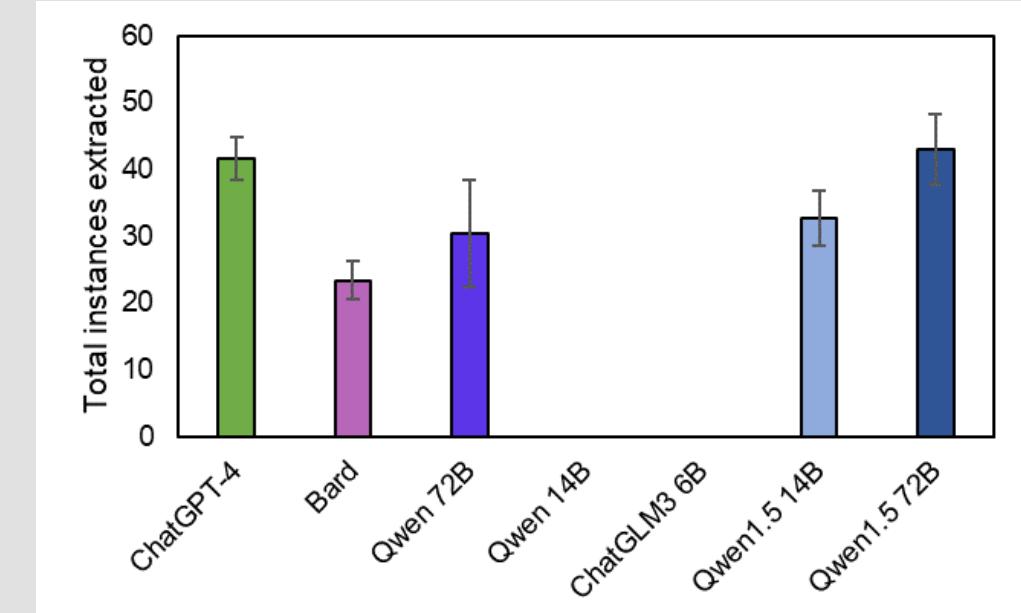
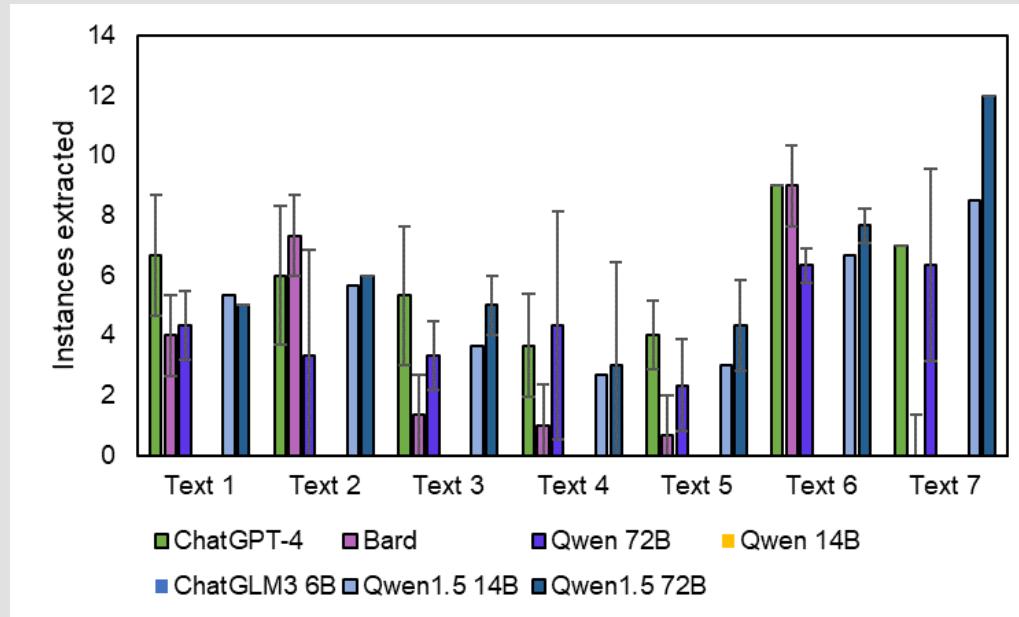


- Models with small parameter cannot perform this RAG task.
- With proper alignment, 14B can be a good model size, balancing speed and quality.
- Closed-source models are reluctant to produce long answers.



# Result: compare LLM information retrieval task

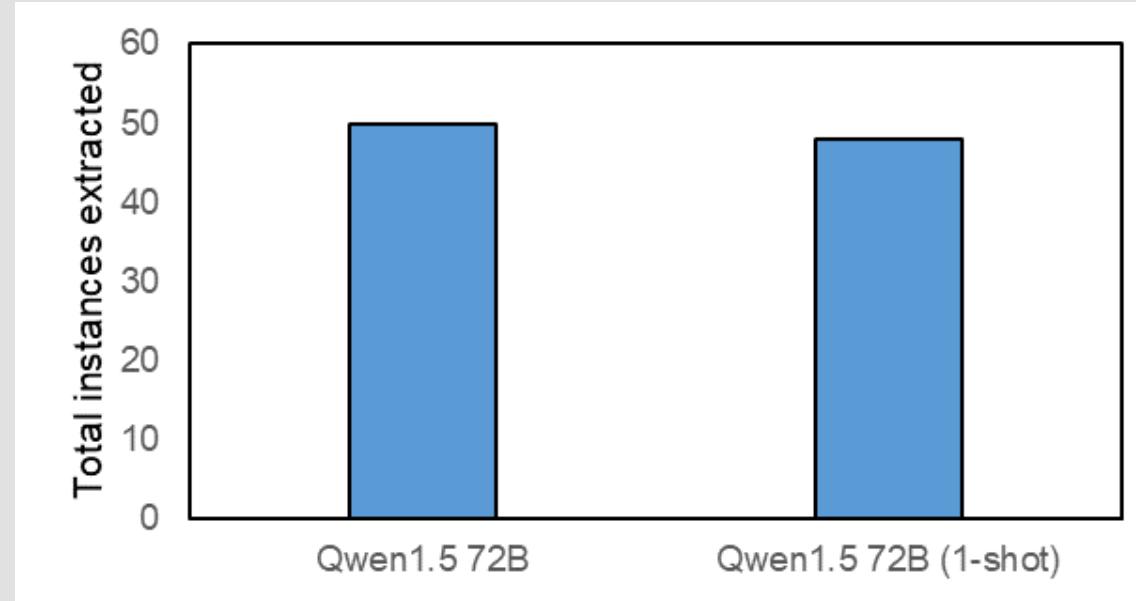
Data extracted in each text, average and standard deviation



- LLM has variations between trials, but not too large variations.
- GPT-4 has the smallest standard deviation between trials.



# Result: Does few shot help LLM?

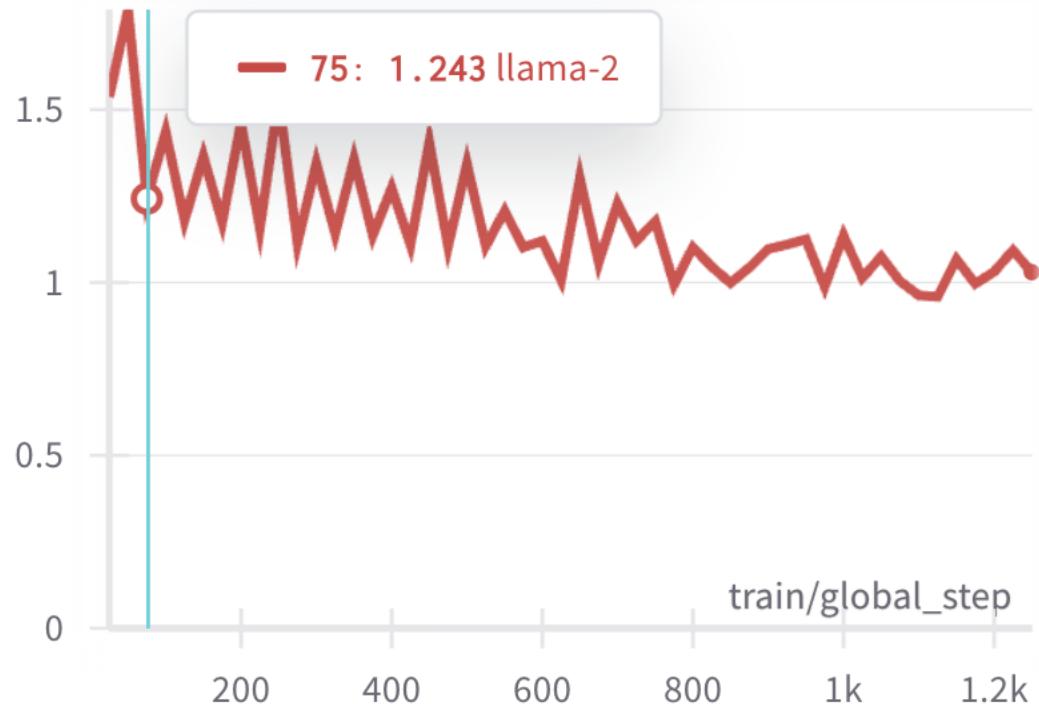


- When LLM already has decent performance, few shot hurts the model's reasoning ability.
- Possible reasons: limited parameters, reduced context length, improper structure of prompt, the example is irrelevant to the text given ...



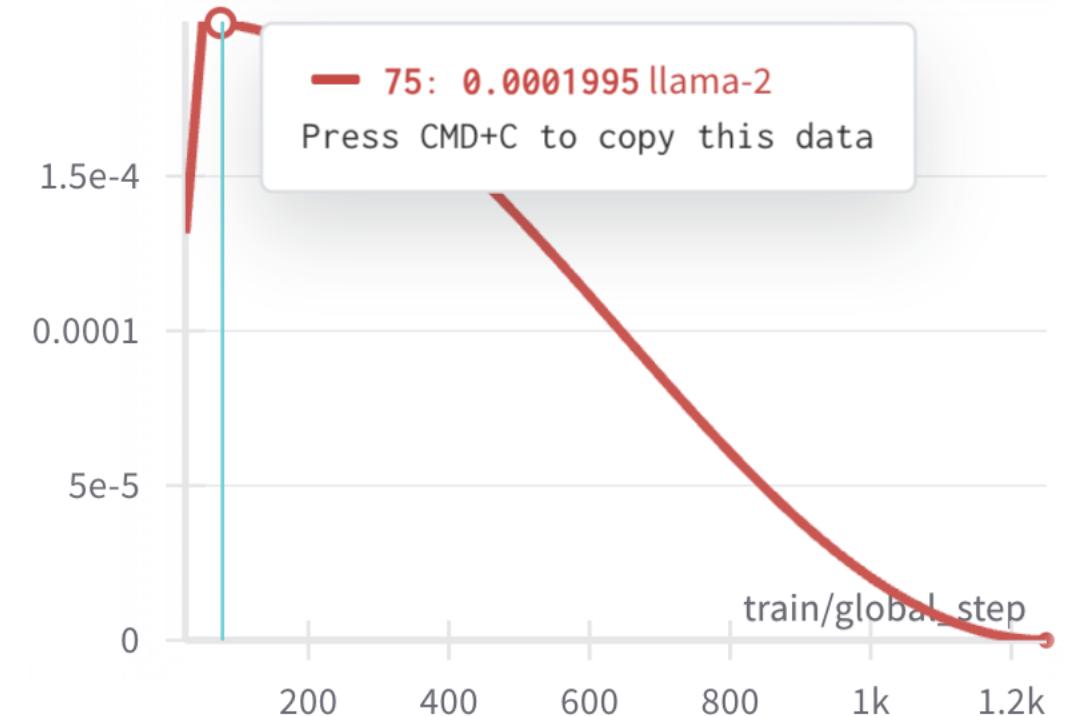
### train/loss

— llama-2

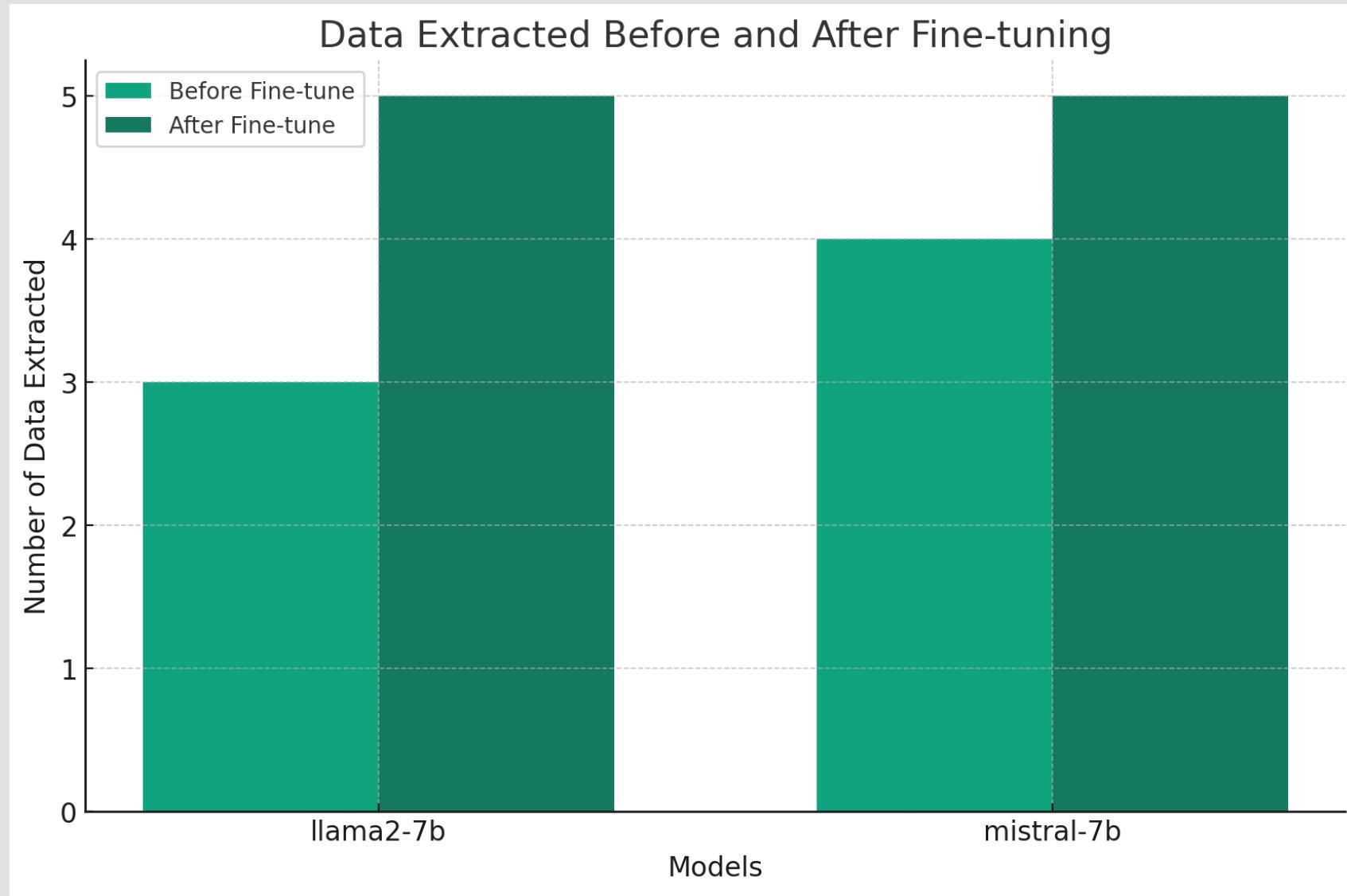


### train/learning\_rate

— llama-2



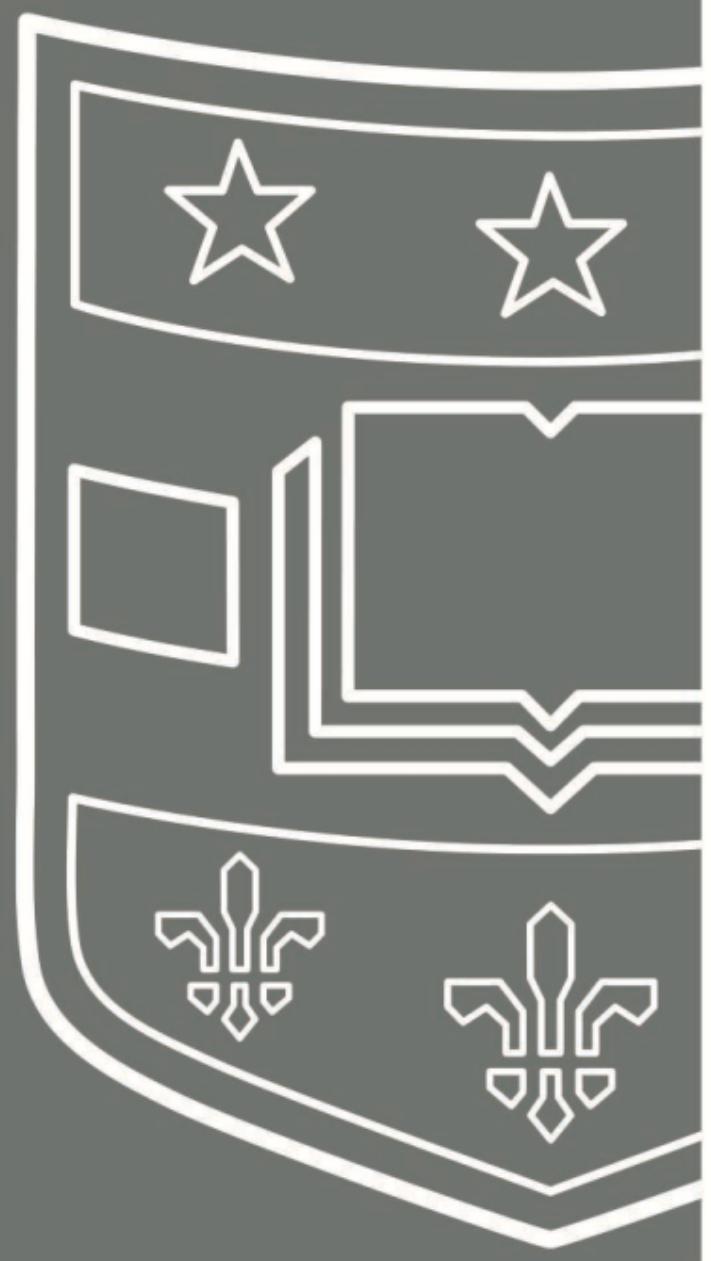
# Paragraph chosen from “method” section



Thank you! Questions?

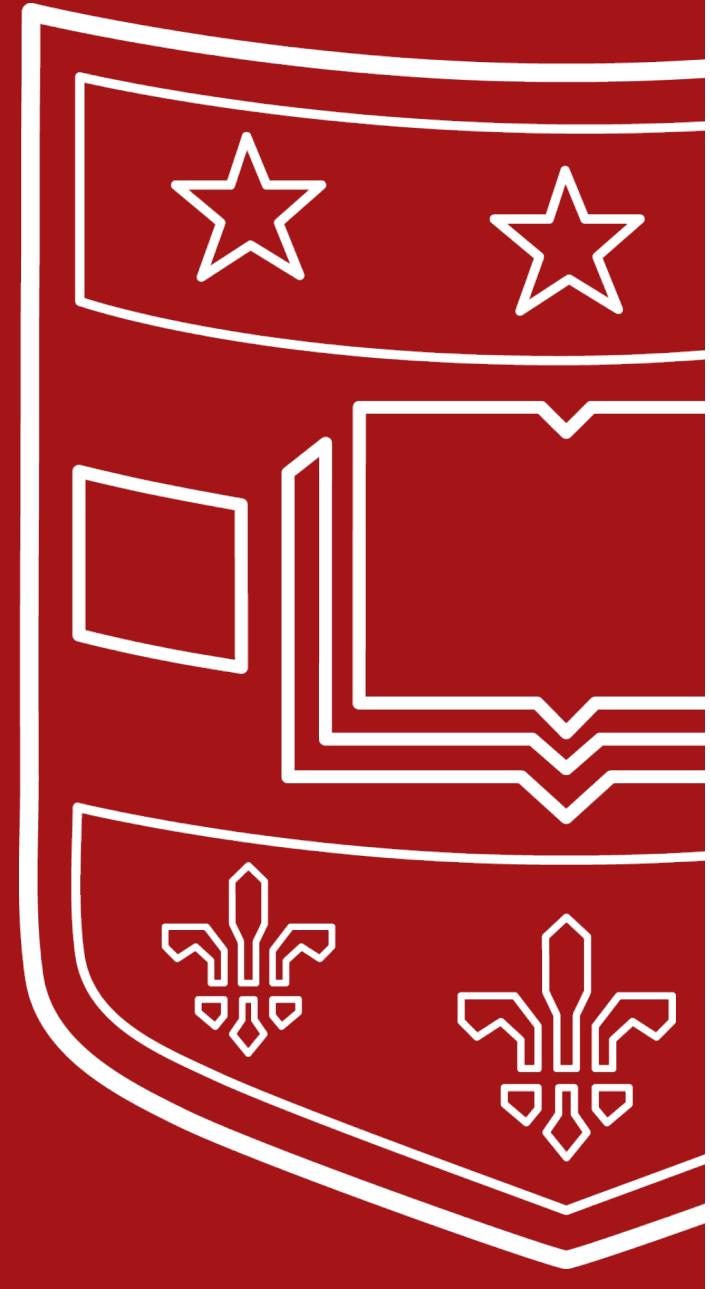


STUDENT AFFAIRS



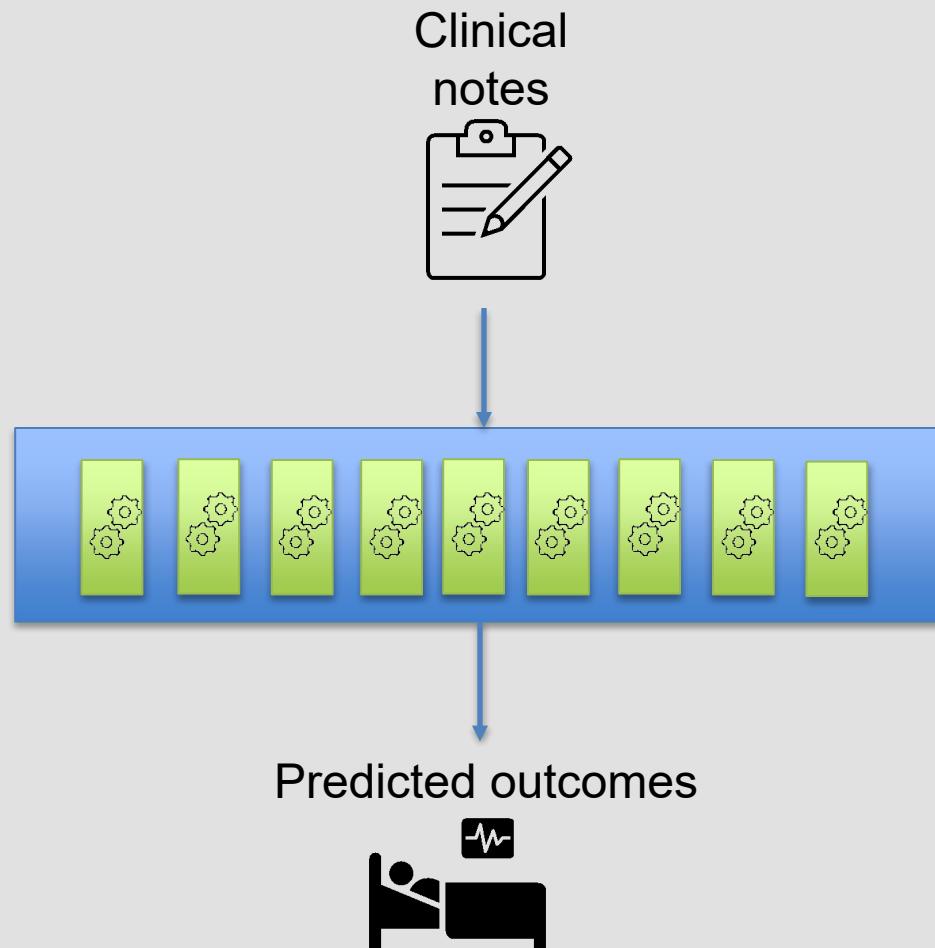
# Exploiting tabular data for more informed initialization in Parameter-efficient fine-tuning methods

*By Charles Alba, Anxu Wang, Hangyue Li*





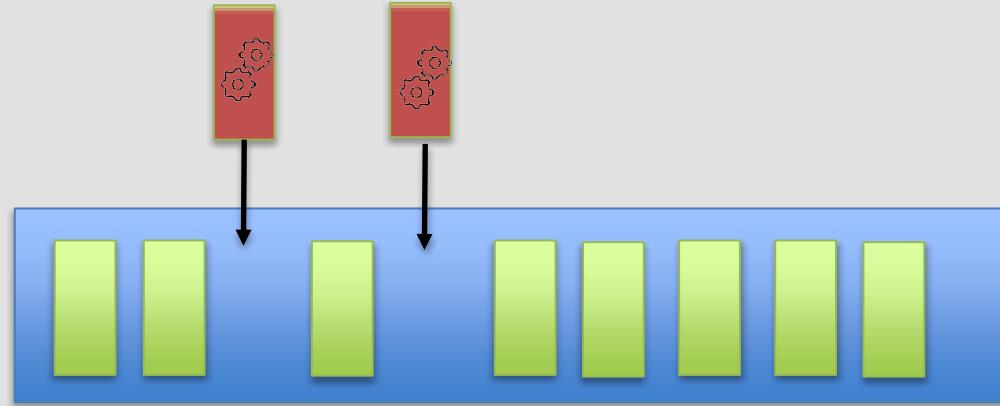
# Fine-tuning





# Types of Parameter Efficient Finetuning: Adapter-based methods

- Adapter  
(Houlsby et. al, 2019)
- Compacter  
(Mahabadi, 2021)



We tune these inserted parameters instead!

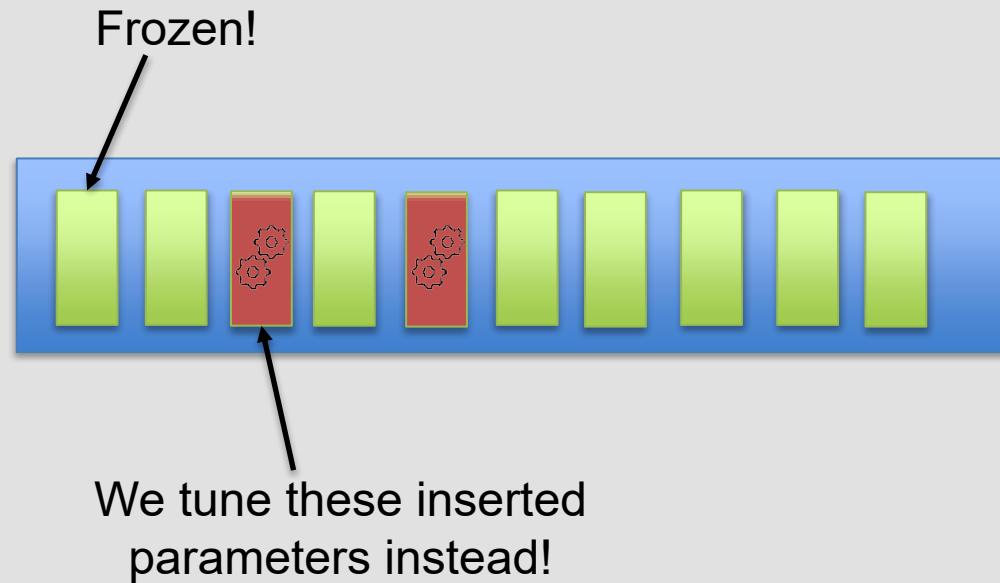
1. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

2. Karimi Mahabadi, R., Henderson, J., & Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34, 1022-1035.



# Types of Parameter Efficient Finetuning: Adapter-based methods

- Adapter  
(Houlsby et. al, 2019)
- Compacter  
(Mahabadi, 2021)



1. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., ... & Gelly, S. (2019, May). Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning* (pp. 2790-2799). PMLR.

2. Karimi Mahabadi, R., Henderson, J., & Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34, 1022-1035.



# Addition-based methods

- Prompt tuning  
(Lester et al, 2021)
- Prefix Tuning  
(Li et al, 2021)
- P-tuning  
(Liu et al, 2021)

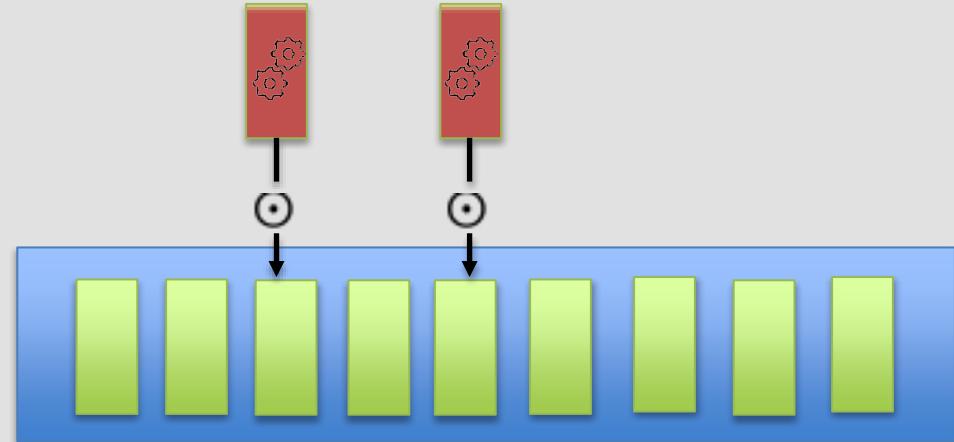


1. Lester, B., Al-Rfou, R., & Constant, N. (2021, November). The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 3045-3059).
2. Li, X. L., & Liang, P. (2021, August). Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 4582-4597).
3. Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., & Tang, J. (2022, May). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 61-68).

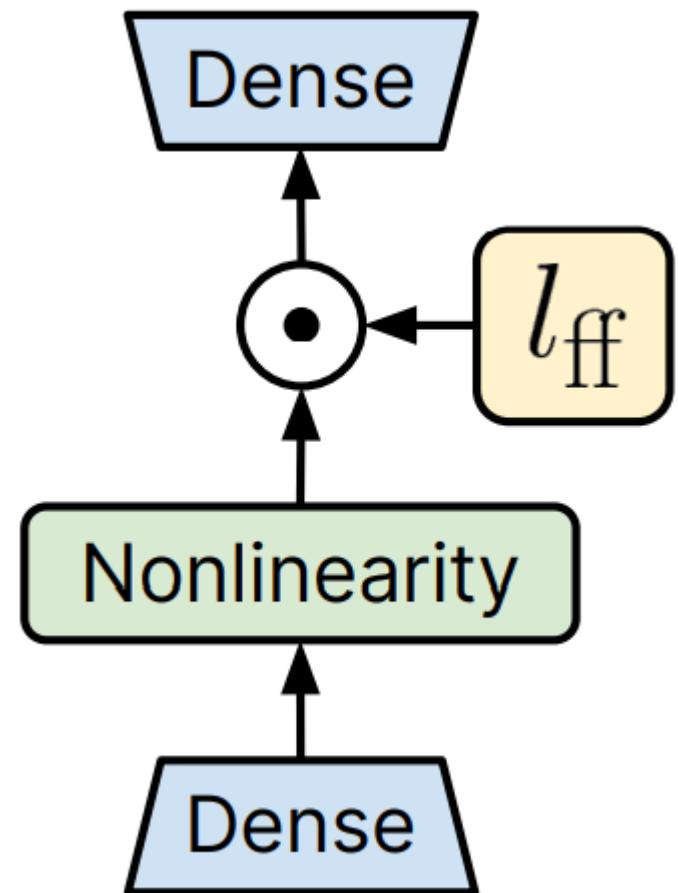
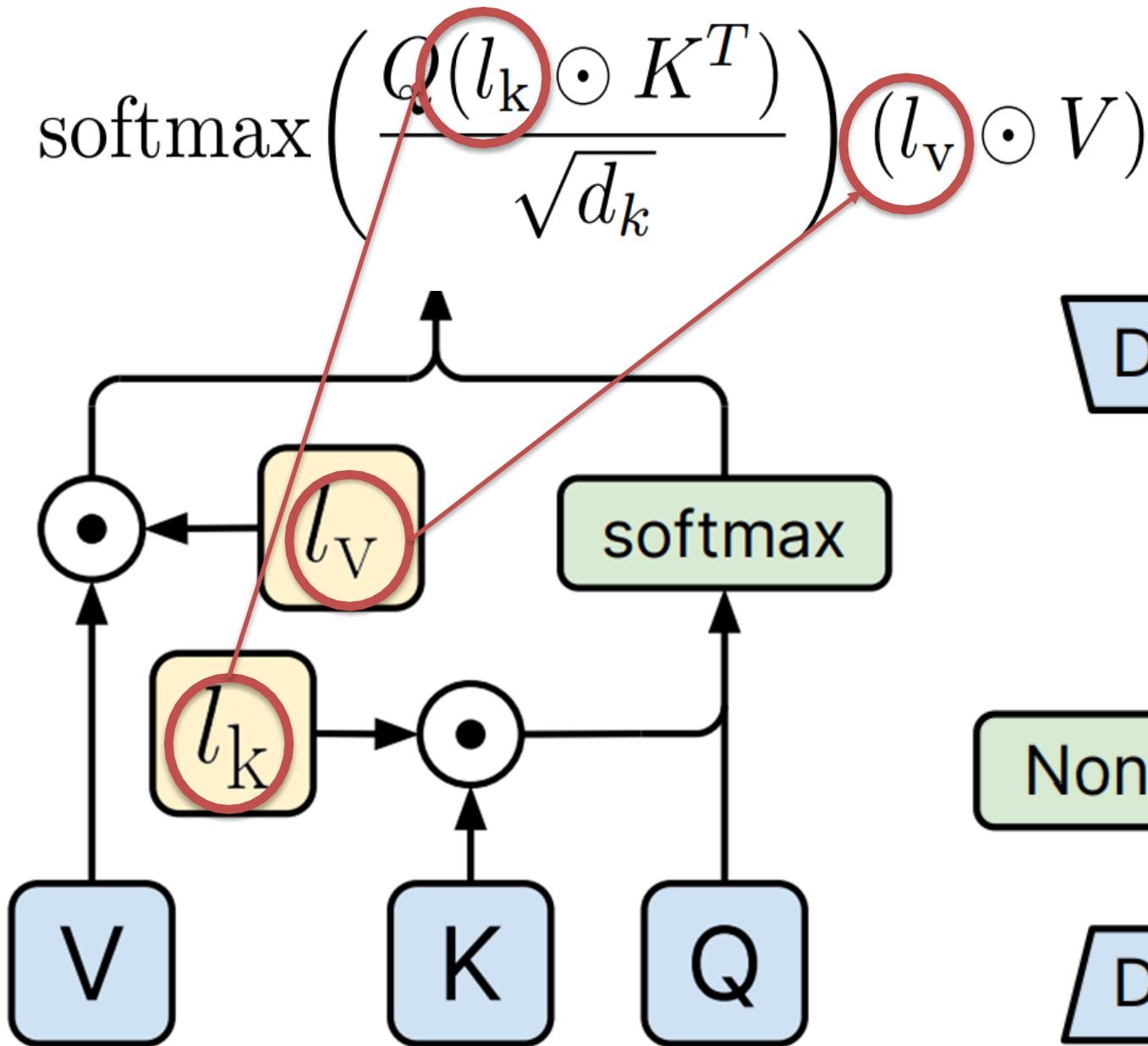


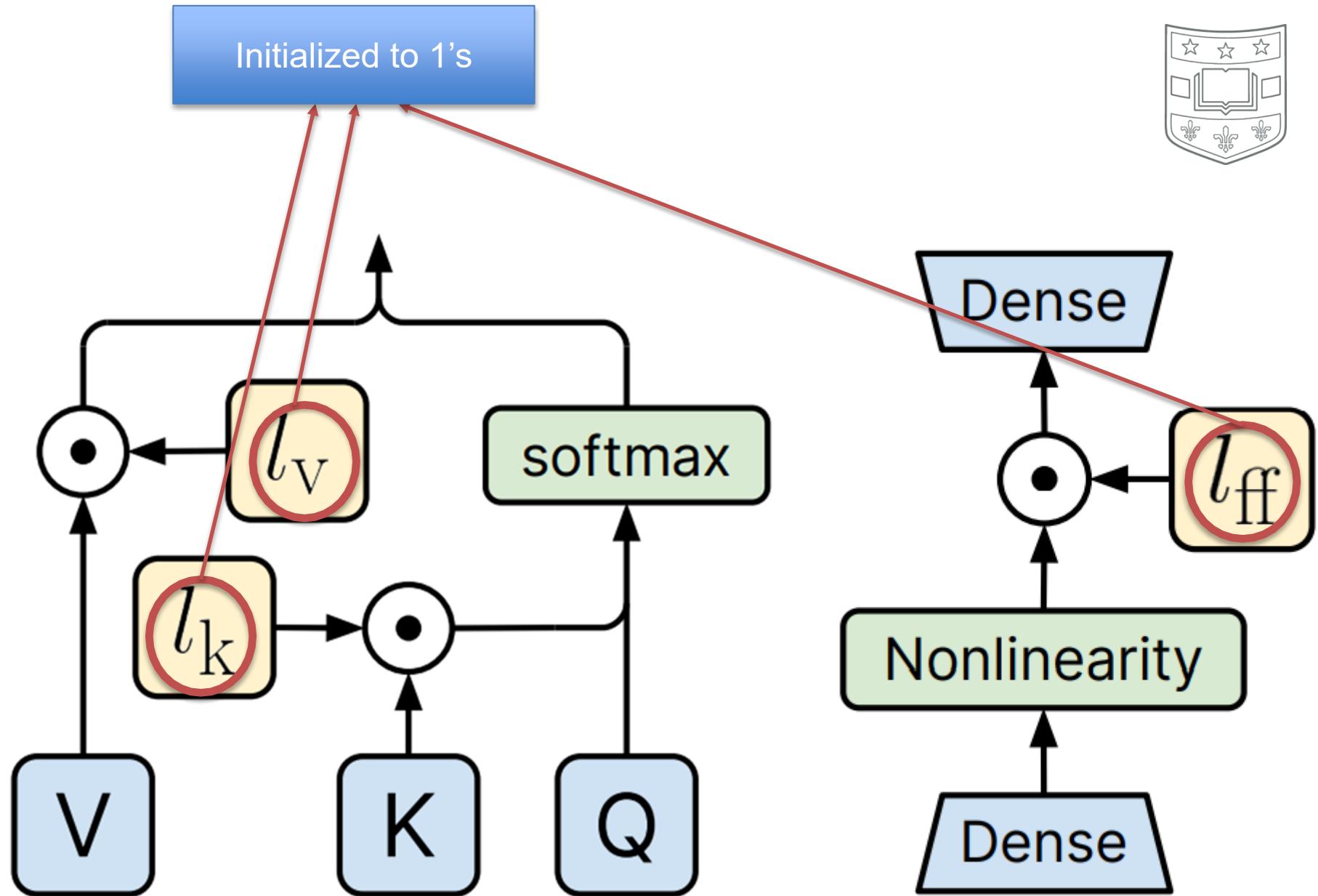
# Reparameterization methods

- LoRa (Hu et al, 2021)
- IA3 (Liu et al, 2023)

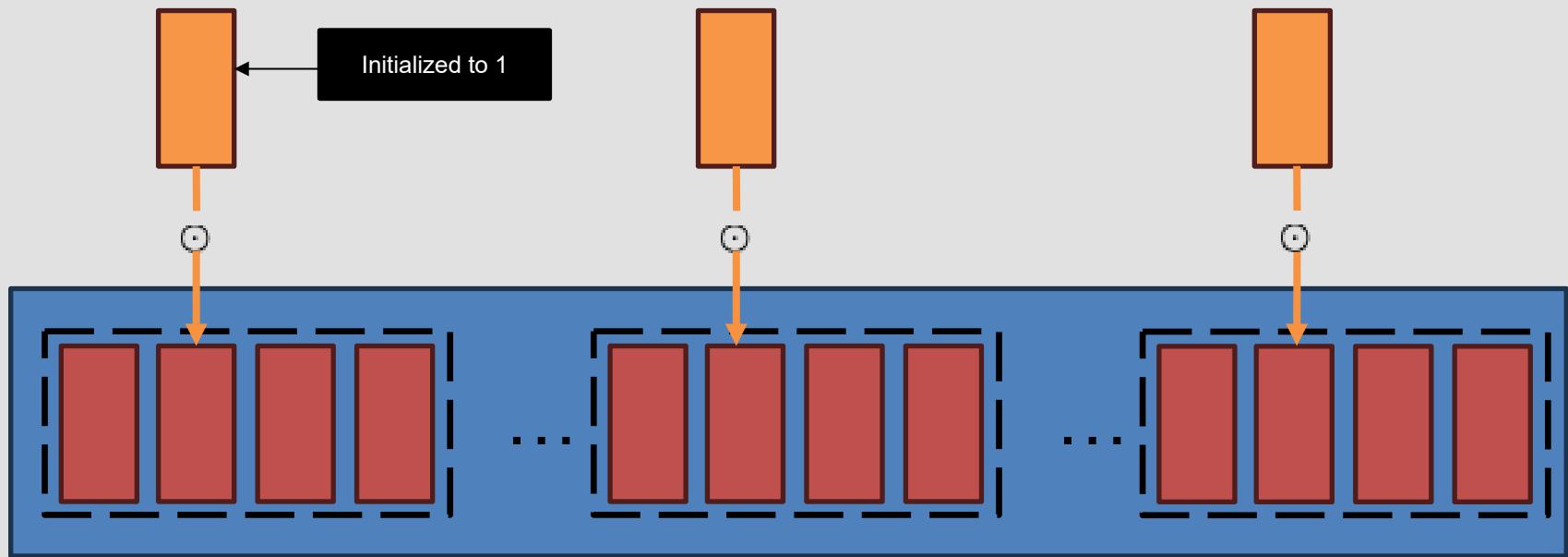


1. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
2. Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., & Raffel, C. A. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35, 1950-1965.





# Problem 1: Is there a better way to initialize IA3 than just with 1's?

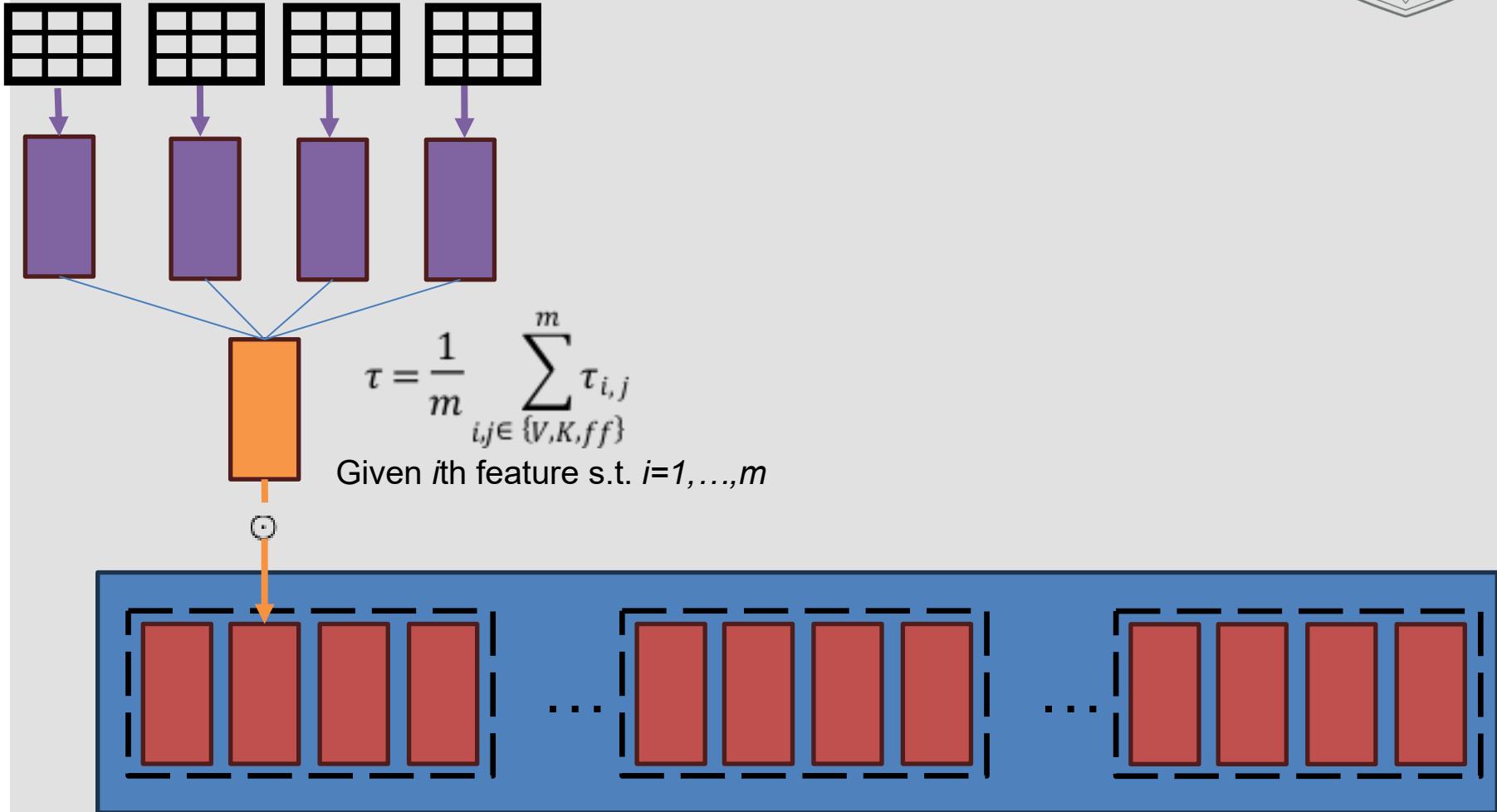




# Problem 2

- In machine learning for medical care, tabular features and textual features are often regarded as separate entities for clinical predictions
- Tabular data holds important information but is never used in the fine-tuning process of LLMs

# Our solution: Incorporate tabular features as part of IA3's initialization process





# Experiment set-up

## Data: MIMIC-IV

- Patient information from the Beth Israel Deaconess Medical Center in Boston, MA
- These include clinical notes, tabular features and patient outcomes
- Publicly available





# Experiment set-up

## Textual Features

- Clinical Notes

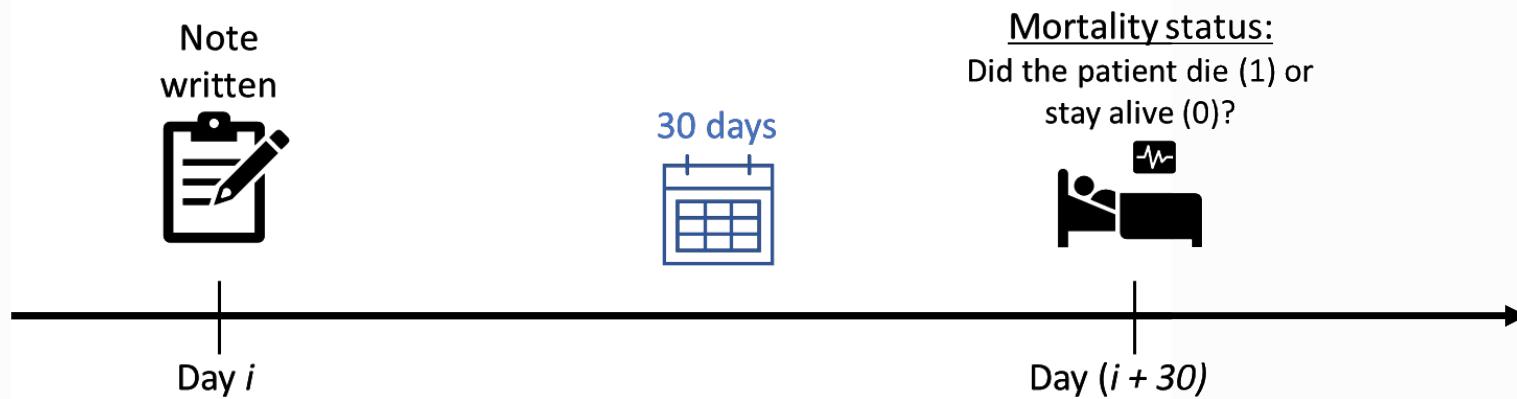
## Tabular features

- Race
- Marital Status
- Language
- Insurance type
- Admission location
- Admission type



# Experiment set-up

**Outcome:** Does the patient die within 30 days?



Positive rate: 2.8%



# Experiment set-up

## Model: bio+ClinicalBERT

- Trained on:
  - bioMedical literature from PubMed database
  - Clinical notes from MIMIC-III





# Results

Fine-tuning type	Accuracy	Precision	Recall	F1
No finetuning	0.972	0.000	0.000	0.000
Full fine-tuning	0.972	<b>0.526</b>	0.079	0.138
IA3	0.972	0.506	0.094	0.158
Our proposed method	0.972	0.475	<b>0.105</b>	<b>0.172</b>



# Discussion

- Using aggregated information from complementary tabular features can help serve as a more ‘informed’ initialization for IA3.
- Helps emphasize more important textual features informed from the predictive tabular features
- Discards the not so important textual features.
- Gives IA3 initialization a ‘good sense of direction’ as to how the model should be tuned.

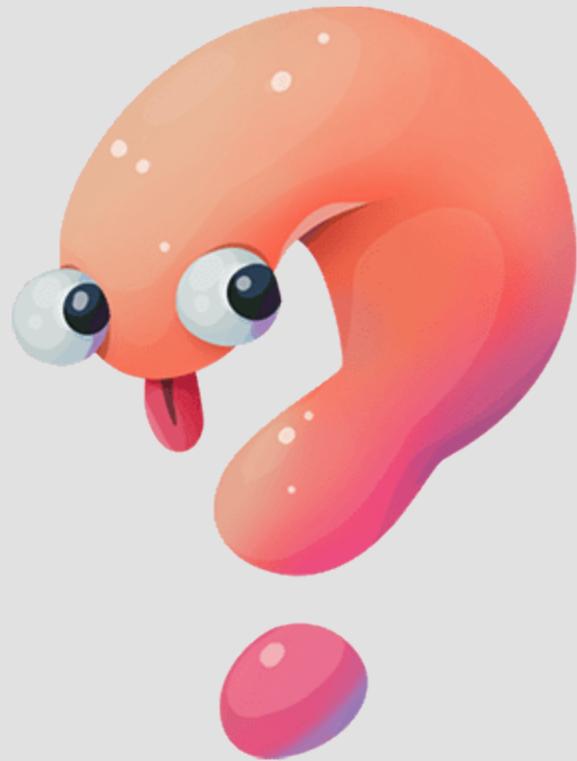


# Thank you





# Questions?



# WrapperAI

---

## *Healthcare's Private Data Sanitizer*

Ver. CSE 561A

# Existing Problem



Businesses are increasingly using advanced tools like **Industry+AI** to enhance their operations.

When uploading files or texts to external services, there is a **significant risk of data leakage**.

**Manually** removing sensitive information is **time-consuming** and **prone to errors**.

# WrapperAI

A **compact** and **efficient automation** model designed to be deployed as a **plugin on an employee's computer**.

Detects and removes sensitive information from documents **without the need for data transfers**.

Allows users to select the information to be removed while **ensuring data security and compliance**.

**Hierarchical Streamlining Control** further optimizes team workflows by structuring tasks and information flow to enhance productivity and efficiency.

# Related Work

## Large Language Models (LLMs)

- Transformer architectures, e.g. BERT, GPT series
- Open-source models, e.g. LLaMA and Vicuna have democratized access

## Fine-Tuning Pre-trained Models

- Allows customization of large models for specific tasks efficiently
- Leverages representations learned on vast data

## Data Protection and Compliance Standards

- HIPAA and GDPR define personal data that needs protection
- Mandate removal/anonymization of specific personal identifiers

## Named Entity Recognition and Data Anonymization

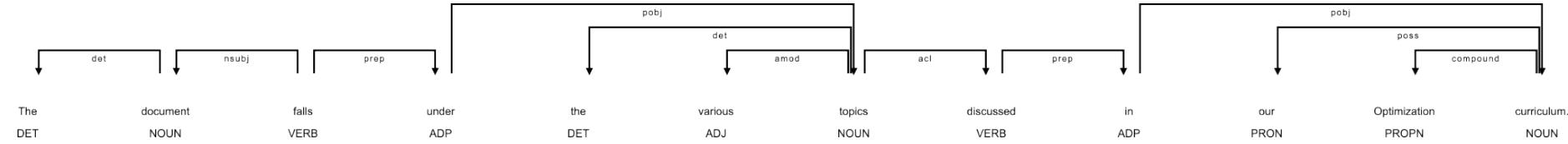
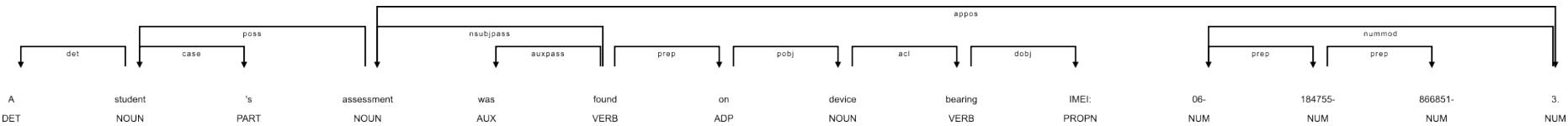
- Blanket anonymization

# NER for Personally Identifiable Information (PII)

	precision	recall	f1-score	accuracy
BERT	0.93	0.95	0.94	0.97
DistilBERT	0.92	0.94	0.93	0.97
RoBERTa	0.96	0.96	0.96	0.97
DeBERTa	0.97	0.98	0.98	0.98

Table 2: Finetuned models' performance on test set

# Sentences



# Learning Linguistic Feature

A	DET	DT	det
student	NOUN	NN	poss
's	PART	POS	case
assessment	NOUN	NN	nsubjpass
was	AUX	VBD	auxpass
found	VERB	VBN	ROOT
on	ADP	IN	prep
device	NOUN	NN	pobj
bearing	VERB	VBG	acl
IMEI	PROPN	NNP	dobj
:	PUNCT	:	punct
06	NUM	CD	nummod
-	SYM	SYM	punct
184755	NUM	CD	prep
-	PUNCT	HYPH	punct
866851	NUM	CD	prep
-	SYM	SYM	punct
3	NUM	CD	appos
.	PUNCT	.	punct
The	DET	DT	det
document	NOUN	NN	nsubj
falls	VERB	VBZ	ROOT
under	ADP	IN	prep
the	DET	DT	det
various	ADJ	JJ	amod
topics	NOUN	NNS	pobj
discussed	VERB	VBN	acl
in	ADP	IN	prep
our	PRON	PRP\$	poss
Optimization	PROPN	NNP	compound
curriculum	NOUN	NN	pobj
.	PUNCT	.	punct
Can	AUX	MD	aux
you	PRON	PRP	nsubj
please	INTJ	UH	intj
collect	VERB	VB	ROOT
it	PRON	PRP	dobj
?	PUNCT	.	punct

# Methods

---

## Algorithm 1: Linguistic Features Analysis and NER

---

**Input:** input document  $D$ , finetuned NER model  $\mathcal{M}$ , spaCy NLP model for linguistic annotations  $\mathcal{C}$ , threshold score  $h$   
**Output:** list of tokens that are classified or highlighted  $\mathcal{L}$

```
1:  $Doc \leftarrow \mathcal{C}(D)$ 
2:  $\mathcal{L} \leftarrow \text{LingFeatAnalyzer}(Doc)$ 
3: for  $t^*$  in  $\mathcal{L}$  do
4:   if  $t^*$  label score  $< h$  then
5:     highlight  $t^*$ 
6:   else
7:      $t^* \leftarrow \mathcal{M}(t^*)$ 
8:   end if
9: end for
10: return  $\mathcal{L}$ 
```

---

---

## Algorithm 2: LingFeatAnalyzer

---

**Input:**  $Doc$ ; list of predefined domain-specific words  $\mathcal{W}$ , list of dependencies that are meaningful  $\mathcal{N}$   
**Output:** list of tokens that are most likely to be PII  $\mathcal{L}$

```
1:  $\mathcal{L} \leftarrow []$ 
2: for  $t$  in  $Doc$  do
3:   if  $t.pos$  is ‘PROPN’ or  $t.pos$  is ‘NUM’ then
4:     append  $t.text$  to  $\mathcal{L}$ 
5:   end if
6:   if  $t.dep$  in  $\mathcal{N}$  and  $t.head$  in  $\mathcal{W}$  then
7:     append  $t.text$  to  $\mathcal{L}$ 
8:   end if
9:   if  $t.dep$  is ‘appos’ and  $t.head.pos$  is ‘NOUN’ and
     $t.head$  in  $\mathcal{W}$  then
10:    append  $t.text$  to  $\mathcal{L}$ 
11:  end if
12: end for
13: return  $\mathcal{L}$ 
```

---

# Q&A

Thanks for Watching