

OBS-DIFF: ACCURATE PRUNING FOR DIFFUSION MODELS IN ONE-SHOT

000
001
002
003
004
005 **Anonymous authors**
006 Paper under double-blind review
007
008
009
010
011
012
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
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
1914<br

054 1 INTRODUCTION

056 Recent advances in text-to-image generation have been largely driven by large-scale diffusion mod-
 057 els (Rombach et al., 2022; Ramesh et al., 2022; Xie et al., 2024). These models, such as the Stable
 058 Diffusion 3 and 3.5 series (Esser et al., 2024), are capable of producing stunning images from tex-
 059 tual prompts, revolutionizing fields from digital art to content creation. However, their massive
 060 parameter counts—often in the billions (e.g., 8B in Stable Diffusion 3.5-Large)—create prohibitive
 061 computational and memory demands, severely limiting their broader accessibility.

062 To improve the efficiency of the diffusion models, multiple research avenues have been proposed.
 063 One major line of work focuses on accelerating the sampling process by reducing the number of
 064 denoising steps (Song et al., 2021; Lu et al., 2022) or knowledge distillation (Salimans & Ho, 2022;
 065 Sauer et al., 2024). Orthogonal to these efforts, model compression aims to reduce the intrinsic
 066 computational and memory footprint of the model itself. This category includes methods like quan-
 067 tization (He et al., 2023; Li et al., 2023b; Shang et al., 2023; Li et al., 2023a) and pruning, which is
 068 the primary focus of our work.

069 The rapid evolution of diffusion models underscores the severe limitations of existing pruning tech-
 070 niques. Current methods often lack generality (Fang et al., 2023; Li et al., 2023c; Kim et al., 2024),
 071 as they are typically tailored to specific architectures like the U-Net and are not easily adapted
 072 to large-scale, text-to-image diffusion models with diverse structures (e.g., Multimodal Diffusion
 073 Transformer). Moreover, the efficiency gains from pruning are frequently undermined by computa-
 074 tionally expensive requirements, such as the need for gradient information during pruning (Fang
 075 et al., 2023; Zhang et al., 2024b) or a costly post-pruning fine-tuning stage. Furthermore, unstruc-
 076 tured and semi-structured pruning remains largely unexplored for large-scale text-to-image diffu-
 077 sion models. All of these motivate our central research question: **Can we develop a general and**
 078 **training-free pruning framework capable of pruning diffusion models with diverse architec-**
 079 **tures and supporting multiple pruning granularities in a one-shot manner?**

080 In the domain of Large Language Models (LLMs), one-shot and training-free pruning methods like
 081 SparseGPT (Frantar & Alistarh, 2023) and Wanda (Sun et al., 2024) have achieved remarkable suc-
 082 cess. Subsequent SlimGPT (Ling et al., 2024) and SoBP (Wei et al., 2024) further explored training-
 083 free structured pruning. These layer-wise post-training pruning method approaches efficiently com-
 084 press massive models without requiring costly retraining. However, the existing training-free prun-
 085 ing methods from the LLM field, such as SparseGPT, cannot be directly applied to the diffusion
 086 model. This is due to the unique challenges posed by diffusion models: their iterative nature, where
 087 parameters are shared across multiple denoising steps. Furthermore, the complex architectures in-
 088 troduce additional difficulties for pruning.

089 To bridge this gap, we introduce **OBS-Diff**, a novel one-shot, training-free pruning framework de-
 090 signed specifically for large-scale text-to-image diffusion models. Our approach revitalizes the clas-
 091 sic Optimal Brain Surgeon (OBS) (Hassibi et al., 1992) and tailors it to the unique, iterative nature of
 092 the diffusion denoising process. By reformulating the pruning objective to account for the temporal
 093 dynamics of generation and introducing a computationally efficient calibration strategy, OBS-Diff
 094 efficiently removes redundant weights with minimal impact on performance, all without requiring
 095 any training during the pruning process or fine-tuning.

096 Our contributions are summarized as below:

- 097 • We adapt the OBS framework to handle the complex architectures of modern diffusion models,
 098 such as the Multimodal Diffusion Transformer (MMDiT), and demonstrate OBS-Diff versatil-
 099 ity across unstructured, semi-structured (e.g., 2:4 sparsity patterns), and structured pruning (e.g.,
 100 removing entire attention heads or FFN neurons).
- 101 • Recognizing that errors introduced in the early stages of the iterative denoising process have a
 102 compounding effect, we propose a **Timestep-Aware Hessian Construction**. This novel construc-
 103 tion weights the importance of parameters according to their influence across the entire denoising
 104 trajectory, prioritizing the more sensitive early steps through a logarithmic weighting scheme.
- 105 • To overcome the prohibitive cost of sequential calibration in iterative models, we devise a **group-**
 106 **wise sequential pruning** strategy built upon “Module Packages”. This approach amortizes the
 107 expensive data collection process by processing layers in batches, striking an effective balance
 108 between computational time and memory requirements for the pruning process.

108 • Extensive experiments demonstrate that OBS-Diff sets a new state-of-the-art for training-free dif-
 109 fusion model pruning. It achieves inference acceleration while maintaining high visual quality,
 110 outperforming other layer-wise pruning methods across various sparsity levels and patterns.
 111

112 **2 RELATED WORK**
 113

114 **Pruning for Diffusion Models.** Several methods have explored pruning for diffusion models. An
 115 early approach, Diff-pruning (Fang et al., 2023), introduced a gradient-based method for structured
 116 pruning. However, its demonstration on small-scale, non-text-to-image models (e.g., DDPMs) and
 117 its dependency on expensive retraining limit its applicability to modern, large-scale systems with
 118 diverse architectures. A significant line of research has since focused on compressing the UNet-
 119 based text-to-image diffusion models, with works like SnapFusion (Li et al., 2023c), MobileDiff-
 120 fusion (Zhao et al., 2024), BK-SDM (Kim et al., 2024), LAPTOP-Diff (Zhang et al., 2024a), and
 121 LD-Pruner (Castells et al., 2024) all targeting less salient components of the UNet architecture.

122 Other works have explored different architectures or techniques; for instance, Tinyfusion (Fang
 123 et al., 2025) introduced depth pruning for the DiT architecture. More recently, EcoDiff (Zhang
 124 et al., 2024b) introduced a general pruning framework for text-to-image models applicable to di-
 125 verse architectures; however, it remains dependent on a costly training phase to learn a pruning
 126 mask and requires extensive hyperparameter tuning. A common theme among these methods is a
 127 dependency on training or fine-tuning and a primary focus on architecture-specific, structured prun-
 128 ing. Furthermore, unstructured and semi-structured pruning for large-scale, text-to-image diffusion
 129 models remains a largely unexplored area.

130 **Layer-Wise Pruning Methods.** Early post-training compression methods, notably Optimal Brain
 131 Damage (OBD) (LeCun et al., 1989) and Optimal Brain Surgeon (OBS) (Hassibi et al., 1992),
 132 utilized Hessian-based saliency scores to prune individual weights. However, the prohibitive cost
 133 of computing and storing the full Hessian matrix limited their scalability. This challenge spurred
 134 the development of layer-wise approaches such as L-OBS (Dong et al., 2017) and Optimal Brain
 135 Compression (OBC) (Frantar & Alistarh, 2022), which approximate the Hessian locally to make
 136 pruning tractable.

137 As models scaled to billions of parameters, particularly in Large Language Models (LLMs), new
 138 methods emerged. SparseGPT (Frantar & Alistarh, 2023), Wanda (Sun et al., 2024), and DSnoT
 139 (Zhang et al., 2024c) focused on efficient unstructured and semi-structured (N:M pattern) pruning.
 140 Subsequently, SlimGPT (Ling et al., 2024) and SoBP (Wei et al., 2024) extended the OBS methodology
 141 to a structured granularity. The applicability of the OBS framework has also been demonstrated
 142 in other architectures, such as the Mamba model via SparseSSM (Tuo & Wang, 2025). Nevertheless,
 143 this family of compression methods remains unexplored in the field of diffusion models.

144 **3 PRELIMINARIES**
 145

146 **3.1 LAYER-WISE POST-TRAINING PRUNING**

147 Post-training pruning often decomposes the global network compression problem into a series of
 148 independent, layer-wise subproblems (Hubara et al., 2021; Nagel et al., 2020; Aghasi et al., 2017).
 149 For each layer l , the objective is to find a pruned weight matrix $\hat{\mathbf{W}}_l$ that minimizes the output
 150 reconstruction error, given input activations \mathbf{X}_l and a target sparsity S_l . This is formulated as:

$$\operatorname{argmin}_{\hat{\mathbf{W}}_l} \left\| \mathbf{W}_l \mathbf{X}_l - \hat{\mathbf{W}}_l \mathbf{X}_l \right\|_2^2 \quad \text{s.t.} \quad \text{sparsity}(\hat{\mathbf{W}}_l) = S_l, \quad (1)$$

151 where $\|\cdot\|_2^2$ is the squared Euclidean norm. The network is pruned by sequentially solving this
 152 optimization problem for each layer.

153 **3.2 OPTIMAL BRAIN SURGEON FOR LAYER-WISE PRUNING**

154 The Optimal Brain Surgeon (OBS) framework (Hassibi et al., 1992) offers an efficient solution to
 155 the layer-wise problem in Eq. (1). A key insight of OBS is that the ℓ_2 -norm objective allows the
 156 problem to be decoupled into independent subproblems for each row of the weight matrix \mathbf{W}_l .

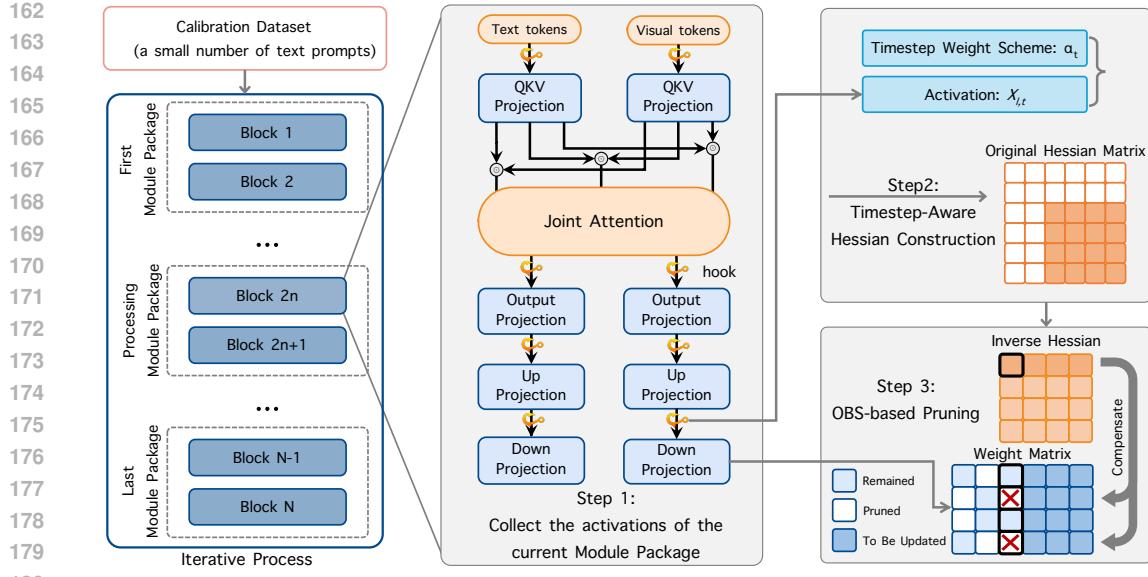


Figure 2: Illustration of the proposed **OBS-Diff** framework applied to the MMDiT architecture. Target modules are first partitioned into a predefined number of “Module Packages” and processed sequentially. For each package, hooks capture layer activations during a forward pass with a calibration dataset. This data, combined with weights from a dedicated timestep weighting scheme, is used to construct Hessian matrices. These matrices guide the Optimal Brain Surgeon (OBS) algorithm to simultaneously prune all layers within the current package before proceeding to the next.

For each row, OBS approximates the objective with a second-order Taylor expansion centered around the current weights. This relies on the Hessian of the reconstruction error, $\mathbf{H} = 2\mathbf{X}_l\mathbf{X}_l^T$. This approximation yields a closed-form solution to identify the least salient weight w_q —the one whose removal minimally increases the error—and to compute the optimal update $\delta\mathbf{w}$ for the remaining weights in its row. The saliency score \mathcal{L}_q and the update are defined as:

$$\mathcal{L}_q = \frac{w_q^2}{2[\mathbf{H}^{-1}]_{qq}}, \quad \delta\mathbf{w} = -\frac{w_q}{[\mathbf{H}^{-1}]_{qq}} \mathbf{H}_{:,q}^{-1}, \quad (2)$$

where $[\mathbf{H}^{-1}]_{qq}$ is the q -th diagonal element of the inverse Hessian and $\mathbf{H}_{:,q}^{-1}$ is its q -th column.

This process is repeated iteratively until the target sparsity S_l is reached. After each weight is removed, the inverse Hessian must be updated. To circumvent the prohibitive cost of full re-inversion and the error accumulation of approximate rank-one updates, SparseGPT (Frantar & Alistarh, 2023) imposes a fixed pruning order. This structural constraint enables efficient and stable updates to the inverse Hessian information using methods like Cholesky decomposition (Frantar et al., 2024) as weights are progressively removed.

4 METHODOLOGY

We propose **OBS-Diff**, a one-shot, training-free pruning framework tailored for diffusion models. As illustrated in Figure 2, our method partitions the model into sequential “Module Packages” to amortize calibration costs. Within each package, we employ a novel Timestep-Aware Hessian construction to prioritize early denoising steps, enabling the simultaneous pruning of all target layers within the current package.

4.1 TIMESTEP-AWARE HESSIAN CONSTRUCTION

The layer-wise pruning objective defined in Eq. (1) is effective for models with a single forward pass, but insufficient for diffusion models, which are iterative and operate over a denoising trajectory

parameterized by discrete timesteps $t \in \{1, \dots, T\}$.¹ The impact of pruning-induced errors is not uniform across this trajectory. Errors introduced in early inference steps (small t) are inherently more damaging, as they propagate and compound through all subsequent steps ($t + 1, \dots, T$), leading to larger deviations in the final output.

Therefore, a robust pruning strategy must prioritize preserving network function during these critical early stages. We reformulate the layer-wise optimization problem to minimize a weighted reconstruction error that places greater importance on earlier, higher-impact steps:

$$\arg \min_{\hat{W}_l} \mathbb{E}_{t \sim [1, T]} \left[\alpha_t \left\| W_l X_{l,t} - \hat{W}_l X_{l,t} \right\|_2^2 \right], \quad (3)$$

Here, $X_{l,t}$ is the input to layer l at step t , and α_t is a step-dependent weight. We define α_t using a simple and effective logarithmically decreasing schedule:

$$\alpha_t = \alpha_{\min} + \frac{\alpha_{\max} - \alpha_{\min}}{\ln(T)} \ln(T - t + 1), \quad t \in \{1, 2, \dots, T\}. \quad (4)$$

This schedule ensures the weight is highest at the beginning of inference and decays smoothly, such that $\alpha_1 > \alpha_2 > \dots > \alpha_T > 0$.

By incorporating this weighting, we adapt the Optimal Brain Surgeon framework (Hassibi et al., 1992). The Hessian, which captures the second-order information of this weighted loss, is now computed as a weighted sum over all inference steps:

$$H_l = 2 \sum_{t=1}^T \alpha_t \mathbb{E}[X_{l,t} X_{l,t}^T], \quad (5)$$

which is termed as *Timestep-Aware Hessian*. It encapsulates the varying importance of parameters over the generation process. Saliency scores derived from its inverse are thus more sensitive to weights that are critical during the early, formative stages of the denoising process, resulting in a more faithfully pruned model.

4.2 MODULE PACKAGES: A GROUP-WISE SEQUENTIAL PRUNING STRATEGY

Conventional post-training pruning methods, such as SparseGPT (Frantar & Alistarh, 2023), employ a sequential layer-wise calibration. This paradigm is computationally prohibitive for diffusion models, as calibrating each layer necessitates executing a full, multi-step denoising trajectory. To address this bottleneck, we introduce **Module Packages**, a group-wise strategy that amortizes calibration costs by processing layers in batches.

Our approach is built upon two concepts. A **Basic Unit** is a set of layers with mutually independent inputs in a forward pass (e.g., query, key, and value projections), allowing for parallel processing. A **Module Package** comprises one or more Basic Units, which are pruned and calibrated collectively. Our framework processes these packages sequentially. For each package, we first execute a *Group-wise Data Collection* phase: we run the complete denoising trajectory once across the calibration dataset, using forward hooks to concurrently gather input statistics for all modules within the package. Subsequently, all modules are pruned simultaneously using their respective Timestep-Aware Hessian matrices.

Crucially, the network state is updated sequentially *between* packages but remains static *within* a package during data collection. This preserves the principle of sequential calibration at a coarser, group-wise granularity, rendering the process computationally feasible. This strategy drastically reduces the number of calibration runs, with the primary trade-off being an increased memory footprint to store multiple Hessian matrices concurrently. Notably, our empirical results demonstrate that pruning accuracy has low sensitivity to package granularity, granting practitioners the flexibility to balance computational cost against memory constraints without a significant performance sacrifice.

4.3 EXTENSION TO SEMI-STRUCTURED AND STRUCTURED PRUNING

A key advantage of our OBS-Diff framework is its adaptability. While focusing on unstructured pruning, it readily extends to both semi-structured and structured sparsity.

¹Here, t denotes the sequential index of the denoising iteration during inference, where T is the total number of inference steps (e.g., for $T = 28$, t ranges from 1, 2, \dots , 28).

270 **Semi-Structured Pruning.** For semi-structured patterns like 2:4 sparsity, the extension is direct.
 271 Within each block of four weights, we simply prune the two with the lowest per-weight OBS-Diff
 272 saliency scores, efficiently creating hardware-friendly models.
 273

274 **Structured Pruning.** For structured pruning of Feed-Forward Network (FFN) layers, we assess a
 275 neuron’s importance by aggregating the saliency of its associated weights. The saliency \mathcal{L}_q for an
 276 entire neuron (column q) and the corresponding weight update are:
 277

$$\mathcal{L}_q = \frac{\sum W_{:,q}^2}{2[\mathbf{H}^{-1}]_{qq}}, \quad \delta \mathbf{W} = -\frac{W_{:,q}}{[\mathbf{H}^{-1}]_{qq}} \mathbf{H}_{:,q}^{-1}, \quad (6)$$

280 where the lowest-scoring neurons are removed.
 281

282 Similarly, for Multi-Head Attention (MHA), we prune entire heads. Our approach, inspired by
 283 SlimGPT (Ling et al., 2024), quantifies the saliency of each head.
 284

285 The calculation begins with the full Hessian matrix, \mathbf{H} , for the output projection layer. For the j -th
 286 head, we consider its weight matrix \mathbf{W}_j and the corresponding Hessian block \mathbf{H}_j . The total saliency
 287 for this head, \mathcal{L}_j , is found by aggregating the importance of its individual weights. The saliency is
 288 calculated as:
 289

$$\mathcal{L}_j = \sum_{k=1}^d \frac{\sum (\mathbf{W}_j)_{:,k}^2}{(\mathbf{H}_j^{-1})_{k,k}}, \quad (7)$$

290 where $(\mathbf{W}_j)_{:,k}$ is the k -th column of the weight matrix \mathbf{W}_j , $(\mathbf{H}_j^{-1})_{k,k}$ is the k -th diagonal element
 291 of the inverse Hessian block, and d is the dimension of each head.
 292

293 However, MMDiT’s joint attention mechanism presents a unique challenge. Shared attention heads
 294 process concatenated multi-modal inputs, but are fed into separate, modality-specific output paths.
 295 This structure yields two distinct importance rankings for the same set of heads (one for each modal-
 296 ity), while OBS-Diff processes the two output projection matrices after separation. To resolve this,
 297 we fuse these rankings into a single, decisive list using Reciprocal Rank Fusion (RRF):
 298

$$S_j^{\text{RRF}} = \frac{1}{k + \text{rank}_A(j)} + \frac{1}{k + \text{rank}_B(j)}, \quad (8)$$

301 where $\text{rank}_A(j)$ is the rank of head j for modality A, and k is a stabilizing hyperparameter (e.g.,
 302 60). This fused score provides a unified ranking to guide the pruning of shared attention heads.
 303

304 Subsequently, the weights of the entire output projection layer are updated using the full Hessian
 305 matrix, \mathbf{H} , following the formulation presented in Eq. (6).
 306

5 EXPERIMENTS

5.1 SETTINGS

310 **Models.** To demonstrate the generalizability of OBS-Diff, we evaluate it across a diverse range
 311 of text-to-image models: Stable Diffusion v2.1-base (866M) (Rombach et al., 2022), Stable
 312 Diffusion 3-Medium (2B) (Esser et al., 2024), Stable Diffusion 3.5-Large (8B), and Flux.1-dev (12B)
 313 (Black Forest Labs, 2024). For comparison with prior work, we also evaluate our method on DDPM
 314 (35.7M) (Ho et al., 2020) trained on the CIFAR-10 (32×32) dataset (Krizhevsky et al., 2009).
 315

316 **Baselines.** For text-to-image models, we compare against methods adapted from the Large
 317 Language Model (LLM) domain for unstructured/semi-structured sparsity, namely Wanda (Sun et al.,
 318 2024) and DSnoT (Zhang et al., 2024c), as well as standard magnitude pruning. For structured prun-
 319 ing, we employ an L1-norm based baseline (Li et al., 2017) and EcoDiff (Zhang et al., 2024b). On
 320 the CIFAR-10 DDPM, our method is directly compared with Diff-Pruning (Fang et al., 2023). The
 321 sparsity refers to the pruning ratio of all the linear layers within MHA and FFN for each MMDiT
 322 block. For calibration, we utilize text prompts from the GCC3M dataset (Sharma et al., 2018). To
 323 ensure a fair comparison, all methods and baselines utilize identical configurations (computational
 324 resources provided in Appendix B).
 325

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
Table 1: Quantitative comparison of unstructured pruning methods on text-to-image diffusion models. The best result per metric is highlighted in **bold**.

(a) SD v2.1-base and SD 3-medium							(b) SD 3.5-large and Flux 1.dev						
Base Model	Sparsity (%)	Method	FID ↓	CLIP ↑	ImageReward ↑	Base Model	Sparsity (%)	Method	FID ↓	CLIP ↑	ImageReward ↑		
SD v2.1-base	40	Dense Model	31.25	0.3142	0.3627	SD 3.5-large	50	Dense Model	31.59	0.3156	0.7549		
		Magnitude	27.86	0.3111	0.1864			Magnitude	35.21	0.3052	0.1465		
		DSnoT	31.63	0.3099	-0.0422			DSnoT	32.82	0.3113	0.2323		
		Wanda	27.96	0.3122	0.1367			Wanda	27.49	0.3123	0.4215		
		OBS-Diff	28.19	0.3131	0.2061			OBS-Diff	29.61	0.3142	0.6146		
	50	Magnitude	49.38	0.2959	-0.5580		60	Magnitude	156.21	0.2302	-2.0296		
		DSnoT	69.05	0.2829	-1.1395			DSnoT	81.99	0.2706	-1.3198		
		Wanda	41.84	0.2988	-0.4704			Wanda	48.80	0.2859	-0.6402		
		OBS-Diff	27.41	0.3102	-0.0356			OBS-Diff	29.15	0.3119	0.3984		
		Dense Model	36.14	0.3162	0.9029			Dense Model	39.16	0.3110	0.9661		
SD 3-medium	50	Magnitude	221.24	0.1864	-2.2719		60	Magnitude	42.06	0.2974	-0.1945		
		DSnoT	63.37	0.2908	-0.5941			DSnoT	41.55	0.3095	0.7111		
		Wanda	43.98	0.3000	-0.1076			Wanda	37.65	0.3086	0.7576		
		OBS-Diff	27.20	0.3167	0.6468			OBS-Diff	39.40	0.3075	0.7777		
		Magnitude	349.53	0.1864	-2.2807		70	Magnitude	251.58	0.2104	-2.2271		
	60	DSnoT	211.58	0.2222	-2.2271			DSnoT	44.35	0.2970	-0.3459		
		Wanda	170.33	0.2352	-2.0641			Wanda	49.68	0.2957	-0.1046		
		OBS-Diff	28.49	0.3099	0.1213			OBS-Diff	39.79	0.2986	0.3697		

The Wanda (Sun et al., 2024) and DSnoT (Zhang et al., 2024c) baselines are originally designed for unstructured and semi-structured pruning of LLMs. Their direct application to diffusion models is non-trivial due to the iterative nature of the diffusion model. Specifically, we extended their pruning logic by incorporating the concept of module packages, enabling them to perform unstructured and semi-structured pruning targeted at the key components of the diffusion architecture. Critically, to ensure an equitable comparison with our Hessian-based method, the adapted DSnoT baseline is configured to use its Hessian-based importance score calculation mode.

Evaluation Metrics. We evaluate the performance of the text-to-image models on a subset of 5K prompts from the MS-COCO 2014 validation set (Lin et al., 2014). The evaluation is based on three metrics: Fréchet Inception Distance (FID) (Heusel et al., 2017), CLIP Score (ViT-B/16) (Hessel et al., 2021), and ImageReward (Xu et al., 2023). For the DDPM on CIFAR-10, we report the FID score. We measure efficiency gains in terms of wall-clock time reduction and the decrease in FLOPs.

5.2 RESULTS OF UNSTRUCTURED PRUNING

The results in Table 1 show the superiority of our OBS-Diff in terms of CLIP score and ImageReward. An interesting phenomenon is observed with the FID metric – the pruned model can occasionally outperform the original dense model. E.g., at 40% sparsity on SD v2.1-base, the *Magnitude* method beats the dense model in FID, while our results suggest *Magnitude* does not produce *visually* better results. It is thereby conceived that FID may not be a very reliable metric here to evaluate different pruning methods.

Regarding the CLIP score, OBS-Diff is the best-performing method in the vast majority of test cases, exhibiting only a slight decrease compared to the dense models. Most notably, OBS-Diff consistently leads in the ImageReward metric across all benchmarks, indicating superior alignment with human aesthetic preferences.

The superiority of our approach becomes most pronounced at high sparsity levels. For example, at 60% sparsity on SD 3.5-Large or 70% on Flux 1.dev, the performance of all baseline methods collapses, resulting in metrics that are significantly worse than ours. This quantitative degradation corresponds to a qualitative failure; as illustrated in Figure 1, the images generated by baseline methods at high sparsity are often totally destroyed and suffer from severe artifacts, whereas OBS-Diff continues to produce high-quality and coherent results. Beyond its performance in generation quality, OBS-Diff is also highly efficient. For instance, the entire pruning process for the 2B-parameter

Table 2: Performance of semi-structured (2:4 sparsity pattern) pruning on the Stable Diffusion 3.5-Large model. Pruning is applied to the 3rd through 25th MMDiT blocks. The best result is shown in **bold**.

Base Model	Method	FID ↓	CLIP ↑	ImageReward ↑
SD 3.5-Large	Dense Model	31.59	0.3156	0.7549
	Magnitude	45.39	0.2945	-0.4705
	DSnoT	32.40	0.3069	0.0307
	Wanda	32.08	0.3036	-0.1363
OBS-Diff	OBS-Diff	32.13	0.3129	0.4493

378
 379
 380
 381
 382
 383 Table 3: Performance of structured pruning on the **SDXL (U-Net)** model across various sparsity
 384 levels. Comparison includes the L1-norm baseline, EcoDiff, and our proposed OBS-Diff. The
 385 **TFLOPs** metric represents the theoretical computational cost for a single forward pass of the entire
 386 UNet. For each sparsity group, the best result per metric is highlighted in **bold**.
 387
 388
 389
 390
 391
 392

Model	Sparsity	Method	#Params	TFLOPs ↓	FID ↓	CLIP Score ↑	ImageReward ↑
SDXL	Dense Model			2.57 B	5.98	29.21	0.3213
	15%	L_1 -norm			71.78	0.3035	-0.0006
		EcoDiff	2.24 B	5.33 (↓10.87%)	34.18	0.3100	-0.1870
		OBS-Diff (Ours)			29.08	0.3215	0.6877
	20%	L_1 -norm			133.07	0.2825	-0.7897
		EcoDiff	2.13 B	5.12 (↓14.38%)	42.98	0.2993	-0.6172
		OBS-Diff (Ours)			29.19	0.3212	0.6461
	30%	L_1 -norm			170.68	0.2711	-1.1694
		EcoDiff	1.91 B	4.70 (↓21.40%)	101.96	0.2465	-1.9161
		OBS-Diff (Ours)			29.75	0.3204	0.4909

393
 394 Table 4: Performance of structured pruning on the Stable Diffusion 3.5-Large model across various
 395 sparsity levels. The first and last transformer blocks were excluded from the pruning process. The
 396 TFLOPs metric represents the theoretical computational cost for a single forward pass of the entire
 397 transformer. For each sparsity group, the best result per metric is highlighted in **bold**.
 398
 399

Base Model	Sparsity (%)	Method	#Params	TFLOPs ↓	FID ↓	CLIP ↑	ImageReward ↑
SD 3.5-Large	Dense Model			8.06 B	11.26	31.59	0.3156
	15%	L_1 -norm			158.89	0.2376	-2.0502
		EcoDiff	7.28 B	9.63 (↓14.5%)	230.97	0.2086	-2.2594
		OBS-Diff			32.64	0.3157	0.6446
	20%	L_1 -norm			189.50	0.2124	-2.2385
		EcoDiff	7.02 B	9.09 (↓19.3%)	293.89	0.2050	-2.2724
		OBS-Diff			32.46	0.3149	0.5475
	25%	L_1 -norm			228.82	0.2040	-2.2651
		EcoDiff	6.76 B	8.55 (↓24.1%)	308.96	0.2037	-2.2686
		OBS-Diff			33.73	0.3128	0.3741
	30%	L_1 -norm			327.48	0.2093	-2.2663
		EcoDiff	6.54 B	8.10 (↓28.1%)	346.38	0.2024	-2.2746
		OBS-Diff			34.51	0.3107	0.2221

412
 413 SD 3-medium model completes in under 15 minutes on a single NVIDIA RTX 4090, highlighting
 414 its excellent cost-effectiveness. Detailed analyses of pruning time and the impact of sparsity on
 415 ImageReward are provided in Appendix C.1.
 416
 417

418 5.3 RESULTS ON SEMI-STRUCTURED PRUNING

419 The results for 2:4 semi-structured pruning are presented in Table 2. Although Wanda obtains
 420 a slightly better FID of 32.08 compared to our 32.13, OBS-Diff shows substantial advantages in
 421 semantic-level metrics. Notably, it surpasses the strongest baseline by a large margin in both CLIP
 422 score (0.3129) and ImageReward (0.4493). This highlights our method’s effectiveness in maintain-
 423 ing high-level semantic consistency and visual fidelity under hardware-friendly sparsity constraints.
 424

425 5.4 RESULTS ON STRUCTURED PRUNING

426 The results are presented in Table 4 and Table 3. The baseline L_1 -norm pruning suffers from catas-
 427 troptic performance degradation even at a modest 15% sparsity, with its FID score deteriorating
 428 from 31.59 to 158.89 on SD 3.5-Large. In stark contrast, our method, OBS-Diff, demonstrates
 429 remarkable resilience. At the same 15% sparsity, OBS-Diff maintains an FID of 32.64, nearly iden-
 430 tical to the dense model’s performance. This robustness persists up to 30% sparsity, where OBS-Diff
 431 sustains a strong FID of 34.51 while the baseline model fails completely (FID of 327.48). These

432
433
434
435
436
Table 6: Ablation study of timestep weighting
strategies, conducted on the SD3-Medium model
at 50% unstructured sparsity. (For reference, the
ImageReward of uniform strategy is 0.6355.)

437 Weight strategy	438 ImageReward \uparrow
439 Linear increase	0.6174
440 Linear decrease	0.6384
441 Log increase	0.6244
442 Log decrease	0.6438

436
Table 7: Ablation study on the impact of the
437 number of module packages on resource usage
438 and performance, conducted on SD3-Medium
439 model at 30% unstructured sparsity.

440 Pkgs.	441 Mem. (GB) \downarrow	442 Time (s) \downarrow	443 ImageReward \uparrow
444 1	445 30.67	446 572.20	447 0.8569
448 4	449 24.05	450 896.52	451 0.8442
452 10	453 22.75	454 1539.37	455 0.8429
456 20	457 22.08	458 2594.95	459 0.8564

444 findings highlight OBS-Diff’s superior ability to preserve critical model structures under aggressive
445 structured pruning.

446 To benchmark our method against established techniques, we incorporate the comparison with
447 EcoDiff (Zhang et al., 2024b), a state-of-the-art structured pruning framework for text-to-image
448 diffusion models, directly into the main experiments. As shown in Table 3 and Table 4, while
449 EcoDiff generally outperforms the naive L_1 -norm baseline on SDXL, it still exhibits significant per-
450 formance degradation compared to our method on both tables, especially at higher sparsity levels.
451 For instance, on the U-Net based SDXL model (Table 3), EcoDiff yields an FID of 101.96 at 30%
452 sparsity, whereas OBS-Diff achieves a substantially better FID of 29.75. This confirms that OBS-
453 Diff generalizes effectively across diverse architectures, outperforming baselines on both MMDiT
454 (SD 3.5) and U-Net (SDXL) backbones.

455 Finally, we compare OBS-Diff with Diff-Pruning (Fang et al., 2023), a well-recognized method
456 that leverages gradient information for structured pruning on small class-conditional DDPMs. The
457 detailed results for this specific comparison are deferred to the Appendix C.3, where our method
458 outperforms Diff-Pruning consistently.

460 5.5 WALL-CLOCK TIME COMPARISON

461 To quantify the practical efficiency gains, we
462 measure the wall-clock time for a single for-
463 ward pass through an MMDiT block of the
464 SD3.5-Large model, on a single NVIDIA 4090
465 GPU with batch size 4, resolution 1024×1024 .
466

467 Table 5 shows that both methods effectively re-
468 duce inference latency. The 2:4 semi-structured
469 approach achieves a $1.23 \times$ speedup, while
470 our structured pruning method attains $1.31 \times$
471 speedup at 30% sparsity. These results vali-
472 date the tangible practical acceleration benefits
473 of applying these pruning techniques.

474 5.6 ABLATION STUDY

475 We perform an ablation study to analyze the impact of three key components: (1) the timestep-
476 aware Hessian construction, (2) the number of module packages, and (3) the number of prompts in
477 the calibration dataset. For this study, all variants are evaluated using the ImageReward metric on
478 1,000 prompts from the MS-COCO 2014 validation set.

479
480 **481 Timestep-Aware Hessian Matrix Establishment.** To incorporate temporal information from the
482 diffusion process, we introduce timestep-aware weighting during the Hessian matrix construction.
483 This method assigns a distinct weight to the hooked activations at each timestep. Empirical results
484 demonstrate that assigning greater importance to earlier inference steps yields superior performance.
485 As shown in Table 6, a logarithmic decrease strategy significantly outperforms other weight distri-
486 bution methods.

487 Table 5: Wall-clock inference time (ms) and
488 speedup for a single MMDiT block under various
489 sparsity schemes.

490 Sparsity Type	491 Time (ms)	492 Speedup
Dense	14.36	/
Semi-structured (2:4)	11.71	$1.23 \times$
Structured (15%)	13.96	$1.03 \times$
Structured (20%)	11.95	$1.20 \times$
Structured (25%)	11.17	$1.29 \times$
Structured (30%)	10.99	$1.31 \times$

486
 487 **Module-Package.** The concept of module packages partitions the
 488 model’s layers for layer-wise compression. This approach intro-
 489 duces a critical trade-off between computational resources and time.
 490 Processing the model in more packages reduces peak GPU memory,
 491 as the Hessian matrix for each pruning step is smaller. However, it
 492 proportionally increases the total runtime because the entire cali-
 493 bration dataset must be forwarded for each package. As shown in
 494 our ablation study (Table 7), while the resource trade-off is evident,
 495 the number of packages does not show a clear, predictable rela-
 496 tionship with the final pruned model’s performance. Consequently,
 497 practitioners can select a configuration that best fits their hardware
 498 constraints without sacrificing final model quality.
 499
 500

501 **The Number of the Prompts in the Calibration Dataset.** The size of the calibration dataset
 502 is a critical hyperparameter that directly influences the quality of the approximated Hessian ma-
 503 trix. To find an optimal size, we evaluated post-pruning performance against the number of text
 504 prompts in the calibration dataset, as shown in Figure 3. The pruned model’s ImageReward score
 505 improves sharply up to 100 prompts and then plateaus, indicating a point of diminishing returns
 506 where additional data offers no significant benefit to the Hessian approximation. Therefore, to bal-
 507 ance performance gains with computational efficiency, we selected 100 prompts for our calibration
 508 dataset in all main experiments.
 509
 510

6 CONCLUSION

511 This work introduces OBS-Diff, a novel one-shot, training-free pruning framework tailored for
 512 large-scale text-to-image diffusion models. By revitalizing the classic Optimal Brain Surgeon
 513 method, we address the unique challenges of iterative denoising through our proposed timestep-
 514 aware Hessian construction, which prioritizes critical early-stage generation steps. To overcome
 515 prohibitive calibration costs, we devise a group-wise sequential pruning strategy that effectively bal-
 516 ances memory overhead and computational efficiency. The versatility of our framework extends
 517 across unstructured, semi-structured, and structured pruning, demonstrating its broad applicability.
 518 Extensive empirical results show that OBS-Diff establishes a new state-of-the-art in training-free dif-
 519 fusion model pruning, consistently outperforming existing methods by maintaining high generative
 520 quality, especially at high sparsity regimes.
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539

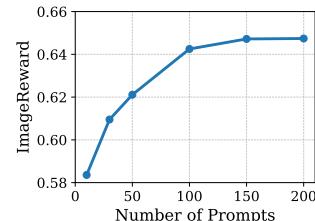


Figure 3: Effect of the number of prompts in calibration dataset on the ImageReward.

540 REFERENCES
541

542 Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. Net-trim: Convex pruning of deep
543 neural networks with performance guarantee. In *NeurIPS*, 2017.

544 Black Forest Labs. Flux. <https://blackforestlabs.ai/>, 2024. Accessed: 2025-09-25.
545

546 Thibault Castells, Hyoung-Kyu Song, Bo-Kyeong Kim, and Shinkook Choi. Ld-pruner: Efficient
547 pruning of latent diffusion models using task-agnostic insights. In *CVPR*, 2024.

548 Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise
549 optimal brain surgeon. In *NeurIPS*, 2017.
550

551 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam
552 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for
553 high-resolution image synthesis. In *ICML*, 2024.

554 Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *NeurIPS*,
555 2023.
556

557 Gongfan Fang, Kunjun Li, Xinyin Ma, and Xinchao Wang. Tinyfusion: Diffusion transformers
558 learned shallow. In *CVPR*, 2025.

559 Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training
560 quantization and pruning. In *NeurIPS*, 2022.
561

562 Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in
563 one-shot. In *ICML*, 2023.

564 Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. Optq: Accurate quantization for
565 generative pre-trained transformers. In *ICLR*, 2024.
566

567 Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network
568 pruning. In *NeurIPS*, 1992.

569 Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-
570 training quantization for diffusion models. In *NeurIPS*, 2023.
571

572 Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A
573 reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

574 Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter.
575 Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*,
576 2017.

577 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*,
578 2020.
579

580 Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training
581 quantization with small calibration sets. In *ICML*, 2021.
582

583 Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight,
584 fast, and cheap version of stable diffusion. In *ECCV*, 2024.

585 Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.
586 2009.
587

588 Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *NeurIPS*, 1989.

589 Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for
590 efficient convnets. In *ICLR*, 2017.
591

592 Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng,
593 Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank component for 4-bit
diffusion models. In *ICLR*, 2023a.

594 Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and
 595 Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *ICCV*, 2023b.
 596

597 Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov,
 598 and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds.
 599 In *NeurIPS*, 2023c.

600 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr
 601 Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
 602

603 Gui Ling, Ziyang Wang, and Qingwen Liu. Slimgpt: Layer-wise structured pruning for large lan-
 604 guage models. In *NeurIPS*, 2024.

605 Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast
 606 ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022.
 607

608 Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up
 609 or down? adaptive rounding for post-training quantization. In *ICML*, 2020.

610 Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-
 611 conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
 612

613 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
 614 resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

615 Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In
 616 *ICLR*, 2022.
 617

618 Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion dis-
 619 tillation. In *ECCV*, 2024.

620 Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on
 621 diffusion models. In *CVPR*, 2023.
 622

623 Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned,
 624 hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

625 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*,
 626 2021.
 627

628 Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach
 629 for large language models. In *ICLR*, 2024.

630 Kaiwen Tuo and Huan Wang. Sparsessm: Efficient selective structured state space models can be
 631 pruned in one-shot. *arXiv preprint arXiv:2506.09613*, 2025.
 632

633 Jiateng Wei, Quan Lu, Ning Jiang, Siqi Li, Jingyang Xiang, Jun Chen, and Yong Liu. Structured
 634 optimal brain pruning for large language models. In *EMNLP*, 2024.

635 Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang
 636 Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffu-
 637 sion transformers. *arXiv preprint arXiv:2410.10629*, 2024.
 638

639 Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao
 640 Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation.
 641 In *NeurIPS*, 2023.

642 Dingkun Zhang, Sijia Li, Chen Chen, Qingsong Xie, and Haonan Lu. Laptop-diff: Layer pruning
 643 and normalized distillation for compressing diffusion models. *arXiv preprint arXiv:2404.11098*,
 644 2024a.
 645

646 Yang Zhang, Er Jin, Yanfei Dong, Ashkan Khakzar, Philip Torr, Johannes Stegmaier, and Kenji
 647 Kawaguchi. Effortless efficiency: Low-cost pruning of diffusion models. *arXiv preprint
 arXiv:2412.02852*, 2024b.

648 Yuxin Zhang, Lirui Zhao, Mingbao Lin, Sun Yunyun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei
 649 Liu, and Rongrong Ji. Dynamic sparse no training: Training-free fine-tuning for sparse llms. In
 650 *ICLR*, 2024c.

651
 652 Yang Zhao, Yanwu Xu, Zhisheng Xiao, Haolin Jia, and Tingbo Hou. Mobiledonfusion: Instant
 653 text-to-image generation on mobile devices. In *ECCV*, 2024.

657 A DECLARATION OF LLM USAGE

658
 659 The use of Large Language Models (LLMs) in this work served two purposes: (1) to aid and polish
 660 the paper writing, and (2) to generate some of the text prompts used by the diffusion model to create
 661 figures that are shown in the paper.

664 B IMPLEMENTATION DETAILS

665
 666 This section provides further details on the experimental setup, including common configurations,
 667 baseline adaptations, and the computational hardware used for our evaluations.

668
 669
 670 **Common Configurations.** To ensure a controlled and fair comparison, all experiments, unless
 671 otherwise specified, adhere to a common set of configurations. For all text-to-image generation
 672 tasks, we set the output resolution to 512×512 pixels to facilitate rapid experimentation across
 673 the diverse and large-scale models. For our method and baselines such as Wanda (Sun et al., 2024)
 674 and DSnoT (Zhang et al., 2024c), we consistently group model parameters into 4 module packages.
 675 Furthermore, a logarithmic decreasing timestep weighting scheme (log decrease) was uniformly
 676 applied across all diffusion models and pruning methods to schedule the pruning process over the
 677 diffusion timesteps.

678
 679 **Computational Resources.** The training of the DDPM on the CIFAR-10 dataset was conducted
 680 on NVIDIA A100 GPUs. For the large text-to-image models, all pruning methods are training-free.
 681 The pruning and evaluation for Stable Diffusion v2.1-base, Stable Diffusion 3-Medium, and Stable
 682 Diffusion 3.5-Large were performed on a single NVIDIA RTX 4090 GPU, each equipped with
 683 48GB of VRAM. Due to its substantial memory footprint, all experiments involving the FLUX.1-
 684 dev model, including its pruning and evaluation, was conducted on a single NVIDIA A100 GPU
 685 with 80GB of VRAM.

687 C MORE EXPERIMENTAL RESULTS

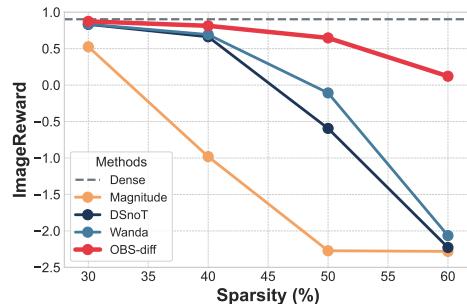
688 C.1 MORE ANALYSIS FOR UNSTRUCTUREDLY PRUNED SD3-MEDIUM

689
 690 As illustrated in Figure 4, our proposed OBS-Diff method consistently outperforms all baseline
 691 approaches in terms of the ImageReward metric across all evaluated sparsity levels. The superiority
 692 of our method is particularly pronounced at higher sparsity ratios. For instance, at 60% sparsity,
 693 the performance of competing methods collapses, yielding negative ImageReward scores. In stark
 694 contrast, OBS-Diff maintains a positive score, demonstrating its exceptional robustness in high-
 695 compression scenarios.

696
 697 In terms of computational efficiency, Table 8 indicates that OBS-Diff has the longest pruning time on
 698 a single NVIDIA RTX 4090. However, the additional overhead is marginal, requiring only slightly
 699 more time than DSnoT (14.95 vs. 14.25 minutes). Considering the substantial gains in genera-
 700 tion quality and model robustness, we conclude that OBS-Diff offers a superior trade-off between
 701 performance and computational cost, establishing it as a highly cost-effective pruning solution.

702
703 Table 8: Pruning time of different unstructured
704 pruning methods on SD3-Medium (2B)
705 at 50% sparsity.

Method	Time (min)
Magnitude	≈ 0
Wanda	7.32
DSnoT	14.25
OBS-Diff	14.95



713
714 Figure 4: ImageReward vs. sparsity for
715 various unstructured pruning methods on SD3-
716 Medium.

717 C.2 STRUCTURED PRUNING FOR SD3-MEDIUM

718
719 We evaluate our structured pruning method, OBS-diff, on the Stable Diffusion 3-medium model
720 and compare it against the widely-used L_1 -norm magnitude pruning baseline. As summarized in
721 Table 9, the baseline method suffers from severe performance degradation as sparsity increases. In
722 contrast, our approach maintains performance remarkably close to the original dense model across
723 all tested sparsity levels, demonstrating its effectiveness and robustness.

724
725 Table 9: Performance comparison of structured pruning methods at 10%, 15%, 20%, and 25%
726 sparsity on the Stable Diffusion 3-medium model (2B). The first and last transformer blocks were
727 excluded from the pruning process. The TFLOPs metric represents the theoretical computational
728 cost for a single forward pass of the entire transformer. For each sparsity group, the best result per
729 metric is highlighted in **bold**.

730 Base Model	731 Sparsity (%)	732 Method	733 #Params	734 TFLOPs ↓	735 FID ↓	736 CLIP ↑	737 ImageReward ↑
		Dense Model	2.03 B	2.84	36.14	0.3162	0.9029
	10%	$L_1 - norm$	1.91 B	2.59 (↓8.8%)	267.32	0.2035	-2.2611
		Ours (OBS-diff)			35.65	0.3166	0.8118
SD 3-medium	15%	$L_1 - norm$	1.83 B	2.43 (↓14.4%)	326.92	0.1942	-2.2768
		Ours (OBS-diff)			34.33	0.3168	0.6717
	20%	$L_1 - norm$	1.78 B	2.31 (↓18.7%)	348.77	0.1926	-2.2768
		Ours (OBS-diff)			33.15	0.3163	0.4997
	25%	$L_1 - norm$	1.72 B	2.19 (↓22.9%)	365.24	0.1906	-2.2786
		Ours (OBS-diff)			32.96	0.3143	0.2782

741 C.3 COMPARISON WITH DIFF-PRUNING ON DDPM

742
743 To evaluate the generalizability of our method beyond large-scale text-to-image models, we adapt
744 it to the task of structured pruning for a Denoising Diffusion Probabilistic Model (DDPM) on the
745 CIFAR-10 dataset. Our adaptation leverages the column masks identified by Diff-Pruning, which
746 are then integrated with our OBS weight update mechanism as detailed in Eq. (6).

747
748 As presented in Table 10, our method surpasses the current state-of-the-art baseline, Diff-Pruning,
749 by achieving a superior FID score under an identical fine-tuning budget (100K steps). This result
750 demonstrates not only the versatility of our approach but also suggests that the model pruned by
751 OBS-Diff serves as a more effective checkpoint for subsequent fine-tuning.

752 D ROBUSTNESS AND GENERALIZATION ANALYSIS

753
754 To demonstrate that our calibration (using only 100 prompts) does not overfit, we evaluated the
755 fixed pruned model (**SD3-Medium, 50% Unstructured**) under inference conditions significantly

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
Table 10: **Performance of pruned DDPMs on CIFAR-10** (32×32). All pruned models are fine-tuned for 100K steps. Evaluations are conducted on samples generated via 100 DDIM steps. The best FID score is highlighted in **bold**.

Method	#Params ↓	MACs ↓	FID ↓	Train Steps ↓
Pretrained	35.7M	6.1G	4.50	800K
Random Pruning	13.95M	2.1G	7.85	100K
Magnitude Pruning	13.95M	2.1G	7.91	100K
Diff-Pruning	13.95M	2.1G	7.72	100K
OBS-Diff	13.95M	2.1G	7.55	100K

different from the calibration settings (CFG 7.0, Steps 25, Euler). For fast evaluation, we evaluated all tasks on the MSCOCO 2014 validation 1K subset using the ImageReward metric.

D.1 ROBUSTNESS TO CFG SCALES

We evaluated the pruned model across varying Classifier-Free Guidance (CFG) scales. As shown in Table 11, the pruned model achieves **higher performance at CFG 9.0 (0.7044)** than at the calibration setting of CFG 7.0 (0.6425). This indicates that our pruning strategy effectively preserves the model’s semantic generation capabilities and generalizes exceptionally well to higher guidance scales, which are critical for high-quality text-to-image synthesis.

Table 11: Robustness of the pruned SD3-Medium (50% Unstructured) across different CFG scales. The model was calibrated at CFG 7.0.

CFG	Dense (ImageReward)	Pruned (ImageReward)	Performance
5.0	0.8319	0.5297	-
7.0 (Calibrated)	0.8510	0.6425	Baseline
9.0	0.8275	0.7044	Improved

D.2 ROBUSTNESS TO SAMPLING STEPS

To address the concern regarding step counts, we evaluated the pruned model (calibrated at 25 steps) across 15, 25, and 50 inference steps. The results are summarized in Table 12. Although calibrated at 25 steps, the pruned model effectively leverages additional compute at 50 steps to generate higher-quality images. This confirms the pruning preserves the integrity of the underlying ODE trajectory.

Table 12: Robustness across varying inference sampling steps. The model was calibrated at 25 steps.

Steps	Dense (ImageReward)	Pruned (ImageReward)	Trend
15	0.6988	0.4883	Fast Preview
25 (Calibrated)	0.8510	0.6425	Baseline
50	0.9391	0.7153	Improved Quality

D.3 ROBUSTNESS ACROSS SAMPLERS

We evaluated generalization across different solvers on both SD3-Medium (MMDiT) and SD v2.1 (U-Net).

- **SD3-Medium:** Calibrated on Euler (1st-order), the model generalizes zero-shot to Heun (2nd-order), showing significant quality gains.
- **SD v2.1:** We applied 40% unstructured pruning (calibrated on PNDM). As shown in Table 13, the relative performance ranking of the samplers is preserved between the Dense

810 and Pruned models (e.g., DPM++ remains the highest performing), indicating the pruning
 811 is solver-agnostic.
 812

813 Table 13: Generalization across different samplers for SD3-Medium and SD v2.1.
 814

815 Model	816 Sampler	817 ImageReward (Dense)	818 ImageReward (Pruned)
817 SD3-Medium	Euler (Calibrated)	0.8510	0.6425
	Heun (2nd Order)	0.9200	0.7249
819 SD v2.1	PNDM (Calibrated)	0.3432	0.1782
	DPM++	0.3889	0.2246
	EDM	0.3442	0.1534
	DDIM	0.3439	0.1579

823
 824 **D.4 GENERALIZATION TO OUT-OF-DISTRIBUTION (OOD) PROMPTS**
 825

826 We address the concern regarding calibration data in two ways:
 827

- 828 **1. Experimental Design:** All main results in the paper already use **GCC3M for calibration**
 829 and **MS-COCO for evaluation**, representing a standard OOD setting.
- 830 **2. New Validation Experiment:** To rigorously test this, we calibrated two separate models—one using **MS-COCO 2014 Train (In-Distribution)** and one using **GCC3M (Out-
 831 of-Distribution)**—and evaluated both on the **MS-COCO 2014 validation 5K subset**.
 832

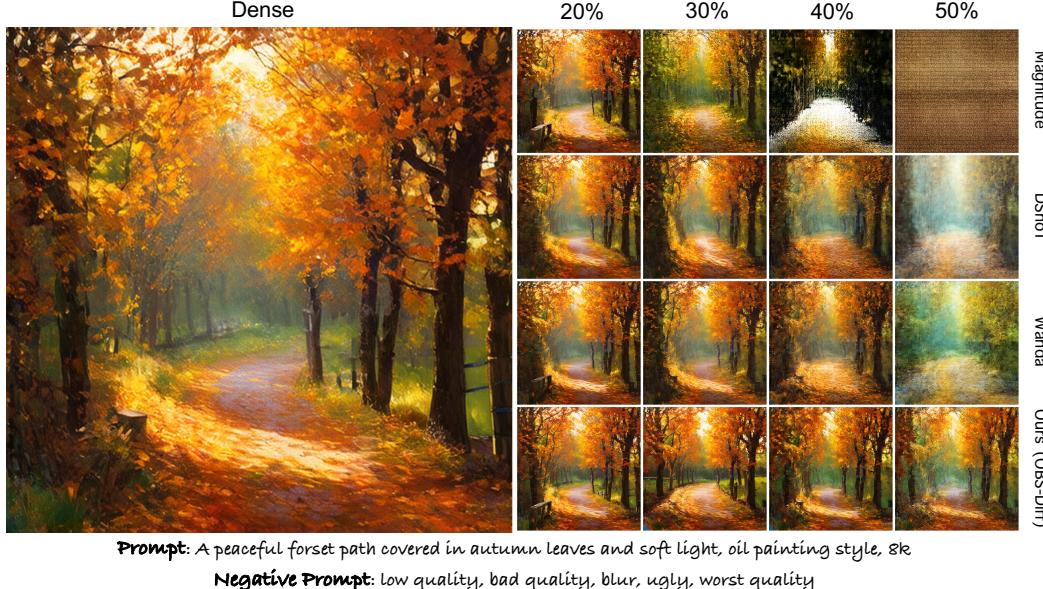
833 As shown in Table 14, the performance is nearly identical. The model calibrated on OOD data
 834 (**GCC3M**) performs on par with (and even slightly better in FID than) the ID model. This definitively
 835 proves that OBS-Diff captures generalizable features and does not overfit to the calibration prompts.
 836

837 Table 14: Comparison of models calibrated on In-Distribution (MS-COCO) vs. Out-of-Distribution
 838 (**GCC3M**) datasets, evaluated on MS-COCO validation set.
 839

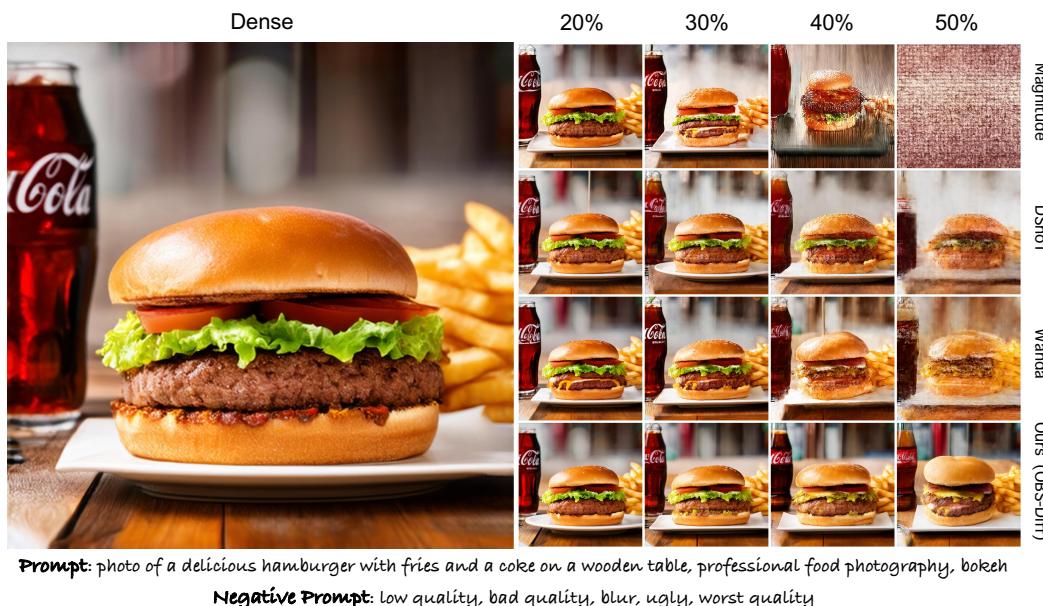
840 Calibration Dataset	841 Evaluation	842 FID ↓	843 CLIP Score ↑	844 ImageReward ↑
842 MS-COCO 2014 Train	In-Distribution (ID)	27.93	0.3169	0.6547
843 GCC3M Train	Out-of-Distribution (OOD)	27.20	0.3167	0.6468

845
 846 **E ADDITIONAL QUALITATIVE RESULTS**
 847

848
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 860
 861
 862
 863



885
886 Figure 5: More Qualitative comparison of unstructured pruning methods on the SD3-Medium
887 model. We evaluate Magnitude, DSnoT, Wanda, and our method (OBS-Diff) at various sparsity
888 levels (20%, 30%, 40%, and 50%) using the same prompt and negative prompt. All images are
889 generated at a resolution of 512×512 .



910
911
912
913 Figure 6: More Qualitative comparison of unstructured pruning methods on the SD3-Medium
914 model. We evaluate Magnitude, DSnoT, Wanda, and our method (OBS-Diff) at various sparsity
915 levels (20%, 30%, 40%, and 50%) using the same prompt and negative prompt. All images are
916 generated at a resolution of 512×512 .

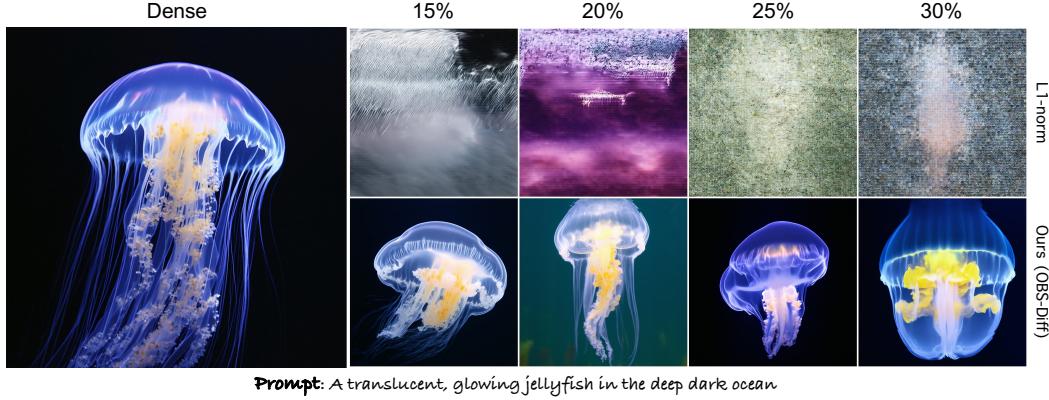


Figure 9: Qualitative comparison of structured pruning methods on the SD3.5-Large model at various sparsity levels (15%, 20%, 25%, and 30%). Results from the L1-norm baseline and our proposed OBS-Diff are shown.

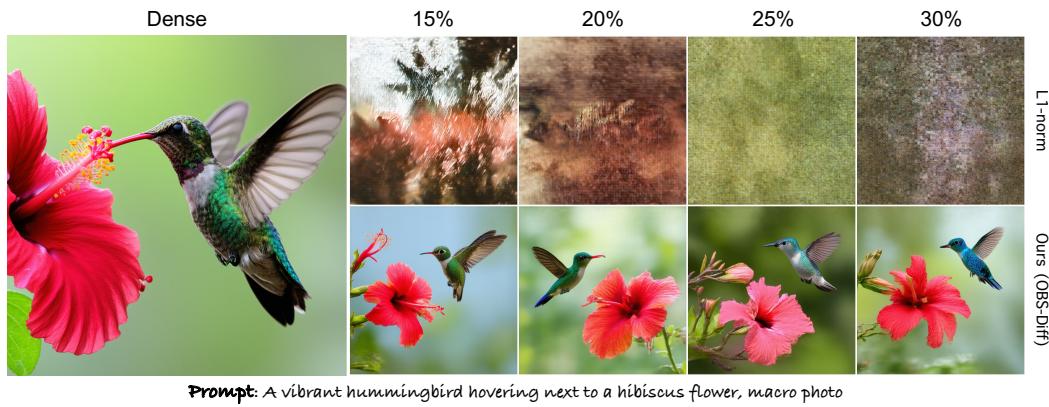


Figure 10: Qualitative comparison of structured pruning methods on the SD3.5-Large model at various sparsity levels (15%, 20%, 25%, and 30%). Results from the L1-norm baseline and our proposed OBS-Diff are shown.



Figure 11: Qualitative comparison of structured pruning methods on the SD3.5-Large model at various sparsity levels (15%, 20%, 25%, and 30%). Results from the L1-norm baseline and our proposed OBS-Diff are shown.