

Scale-space flow for end-to-end optimized video compression

Supplementary Materials

Anonymous CVPR submission

Paper ID 9692

Contents

1. Baselines	1	070
2. Architecture	2	072
3. Calculating the Bjøntegaard Delta Bit Rate	2	074
4. Subjective Comparison	4	076
5. Aggregate Rate-Distortion Graphs	6	077
5.1. UVG (PSNR)	6	078
5.2. UVG (MS-SSIM)	7	079
5.3. MCL-JCV (PSNR)	8	080
5.3.1 Only “Natural” Videos in MCL-JCV (PSNR)	9	081
5.3.2 Only Animated Videos in MCL-JCV (PSNR)	9	082
5.4. MCL-JCV (MS-SSIM)	10	083
5.4.1 Only “Natural” Videos in MCL-JCV (MS-SSIM)	11	084
5.4.2 Only Animated Videos in MCL-JCV (MS-SSIM)	11	085
6. Per-Video Rate-Distortion Graphs	12	087
6.1. UVG	12	088
6.2. MCL-JCV	14	089

1. Baselines

Following prior works, we used *ffmpeg*[1] to produce the evaluation metrics for H.264 and HEVC. Below are the exact commands we used:

H.264 (medium)

```
ffmpeg -i FILE.y4m \
-c:v h264 -crf CRF \
-preset medium -bf 0 FILE.mp4
```

where ‘FILE’ represents the input file and ‘CRF’ the quality level.

HEVC (medium)

```
ffmpeg -i FILE.y4m \
-c:v hevc -crf CRF \
-preset medium \
-x265-params bframes=0 \
FILE.mp4
```

	Encoder	Decoder	
108			162
109	Conv, 5x5/2, 128ch	DeConv, 5x5/2, 128ch	163
110	ReLU	ReLU	164
111	Conv, 5x5/2, 128ch	DeConv, 5x5/2, 128ch	165
112	ReLU	ReLU	166
113	Conv, 5x5/2, 128ch	DeConv, 5x5/2, 128ch	167
114	ReLU	ReLU	168
115	Conv, 5x5/2, 192ch	DeConv, 5x5/2, 3ch	169
116			170

Table 1. The Encoder and Decoder architecture used by the model. Each row denotes a (De)Conv layer or an activation function, where “KxK/s C ch” denotes C output channels and a kernel size of K and a stride of S.

	Hyper Encoder	Hyper Decoder (scale)	Hyper Decoder (mean)	
120				174
121	Conv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	175
122	ReLU	QReLU	ReLU	176
123	Conv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	177
124	ReLU	QReLU	ReLU	178
125	Conv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	DeConv, 5x5/2, 192ch	179
126		QReLU		180
127				181

Table 2. The Hyperprior architecture used by the model. Each row denotes a (De)Conv layer or an activation function, where “KxK/s, C ch” denotes C output channels and a kernel size of K and a stride of S.

where ‘FILE’ represents the input file and ‘CRF’ the quality level.

HEVC (very-fast) Here we used the same settings as reported in [4]:

```
ffmpeg -i FILE.y4m \
-c:v hevc -crf CRF \
-preset veryfast \
-x265-params bframes=0 \
-tune zerolatency \
-x265-params "keyint=12:verbose=1" \
FILE.mp4
```

where ‘FILE’ represents the input file and ‘CRF’ the quality level.

Metric computation For computing frame-level metrics (such as PSNR and MS-SSIM), we dumped the original/encoded videos to frames to sRGB as follows

```
ffmpeg -i FILE FILE_%05d.png
```

We also used this command to obtain the frames that we feed to our models during evaluation.

2. Architecture

In Tables 1&2 we show the full Encoder/Decoder and Hyperprior architecture[?] used in our model. We repurposed this architecture for the ‘Image’, ‘Scale Space Flow’ and ‘Residual’ branches depicted in Figure 2 in the main paper. Following [2] we used QReLU for the Hyperprior to enable deterministic decoding.

3. Calculating the Bjøntegaard Delta Bit Rate

The Bjøntegaard Delta bitrate (BD rate) summarizes the relative performance between two compression models by calculating the relative rate savings for equal quality averaged across the shared quality range. This value is expressed as a percentage [3]. For example, if a codec has a BD rate of 5% relative to H.264, we would expect videos with the same visual quality but which require 5% less space than an H.264 encoding.

216 Due to the typical response of RD curves, a direct calculation of the average percent change between two RD curves will 270
217 weight higher bit rates more than lower bit rates. For this reason, the BD rate is calculated on a log scale for bit rate. The 271
218 original formulation fit a cubic polynomial to exactly four rate points. The area to the left of each curve was then calculated 272
219 via numerical integration. Following common practice, we use a generalized variant of BD rate that relies on piecewise cubic 273
220 hermite interpolating polynomials (PCHIP) for interpolation, which supports a larger number of rate points. Integration uses 274
221 the trapezoid rule after resampling each RD curve at a sufficiently fine granularity (typically 100 points). 275
222 276
223 277
224 278
225 279
226 280
227 281
228 282
229 283
230 284
231 285
232 286
233 287
234 288
235 289
236 290
237 291
238 292
239 293
240 294
241 295
242 296
243 297
244 298
245 299
246 300
247 301
248 302
249 303
250 304
251 305
252 306
253 307
254 308
255 309
256 310
257 311
258 312
259 313
260 314
261 315
262 316
263 317
264 318
265 319
266 320
267 321
268 322
269 323

4. Subjective Comparison



Ours (0.043 bpp)



HEVC (0.0798 bpp)

Figure 1. Comparison between the proposed method and HEVC on a frame from a video where we outperform HEVC. Our reconstruction is slightly sharper while the average bpp is significantly smaller. (Video 'Beauty' in UVG).

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



Ours (0.109 bpp)

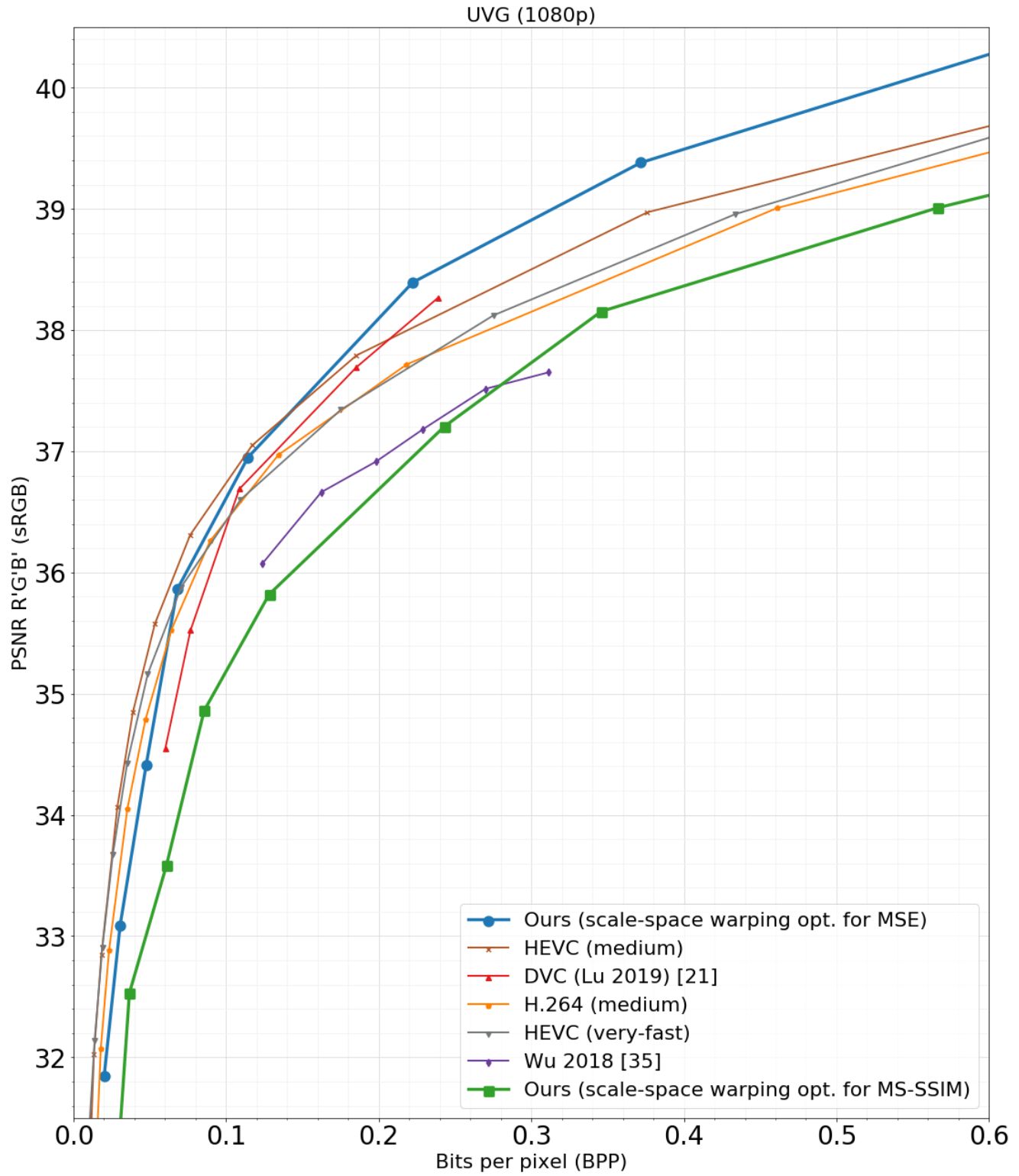


HEVC (0.096 bpp)

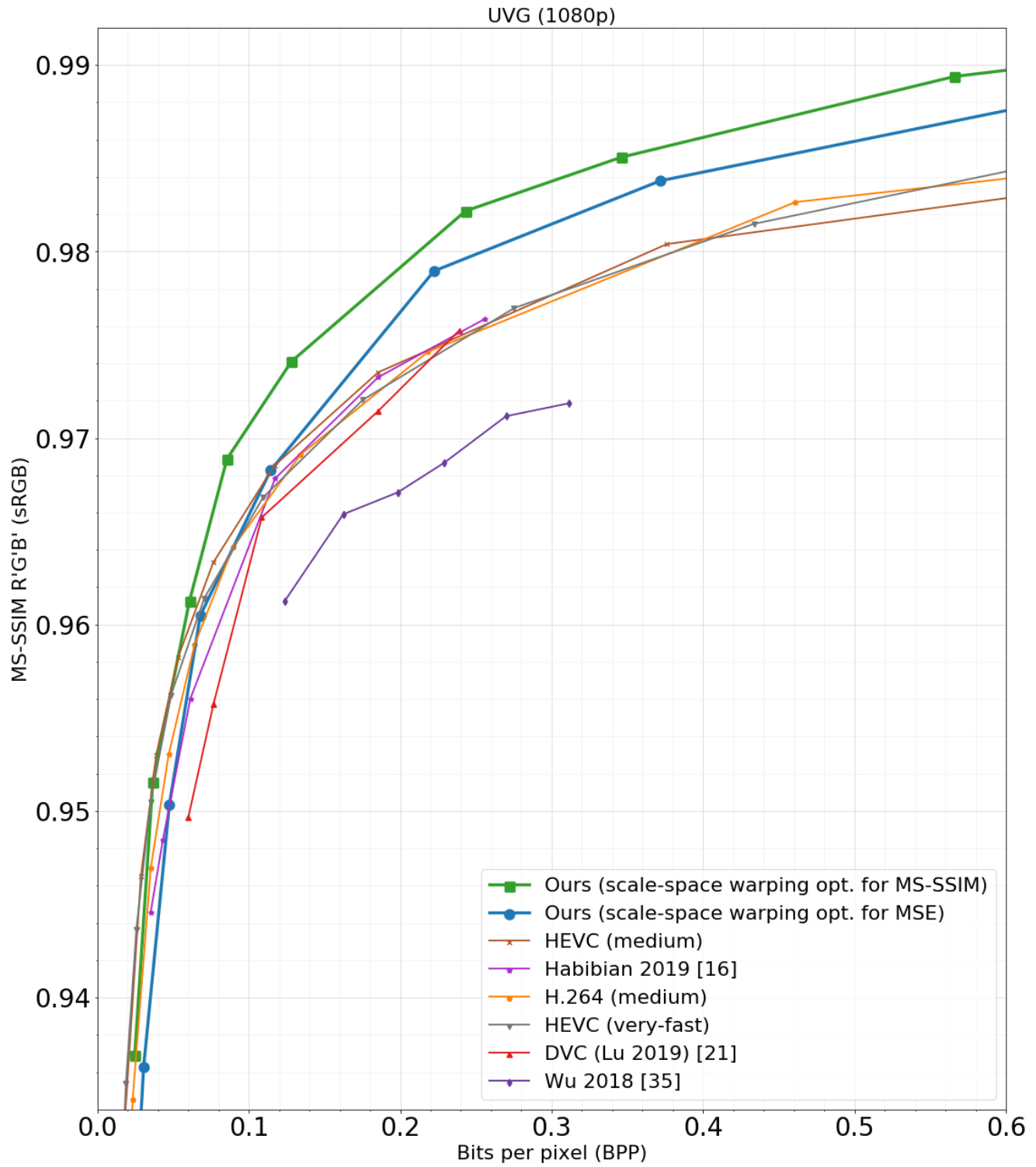
Figure 2. Comparison between the proposed method and HEVC on a frame from an animation, where our method performs poorly according to quantitative metrics. (Video 20 in MCL-JCV).

5. Aggregate Rate-Distortion Graphs

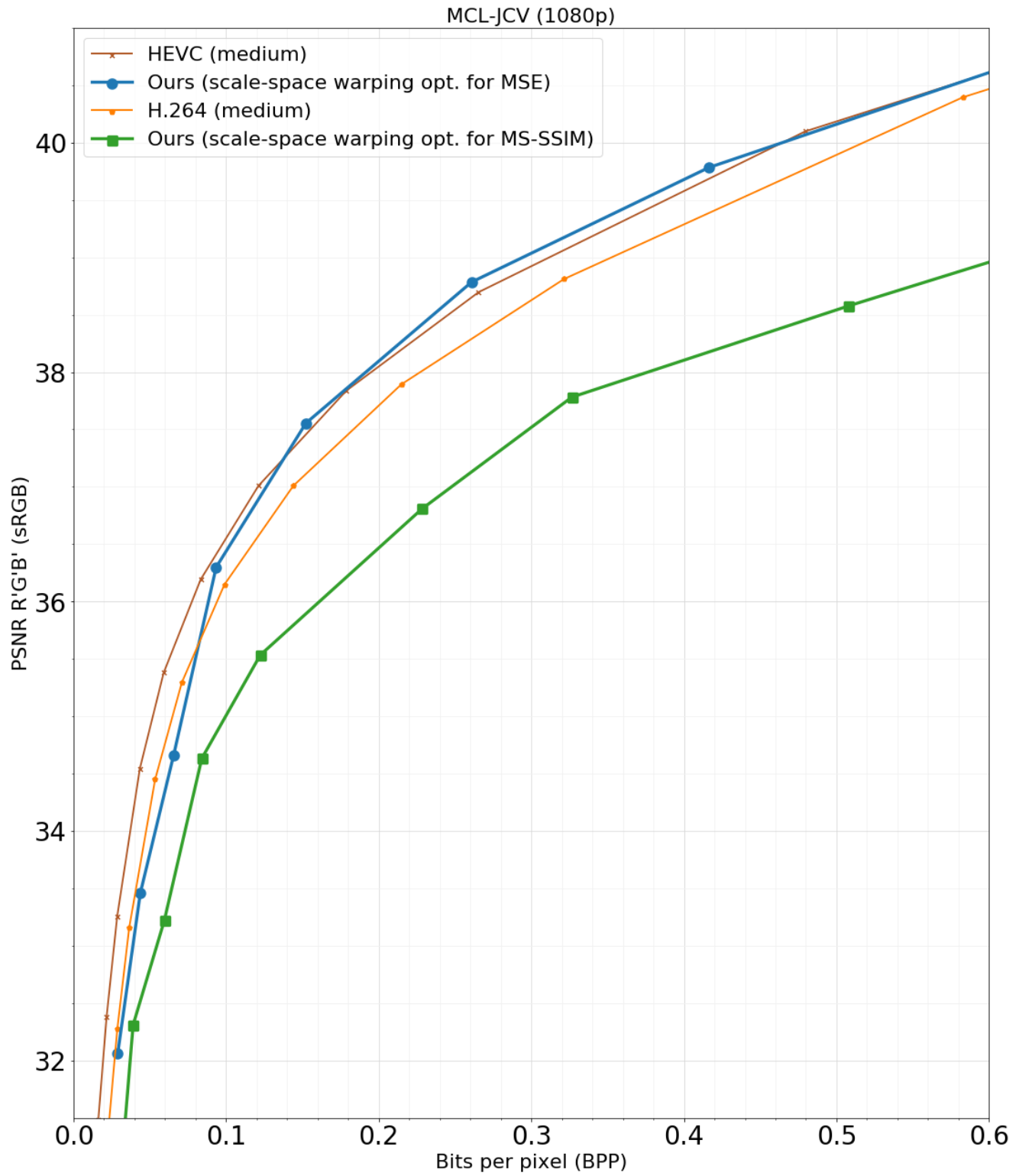
5.1. UVG (PSNR)



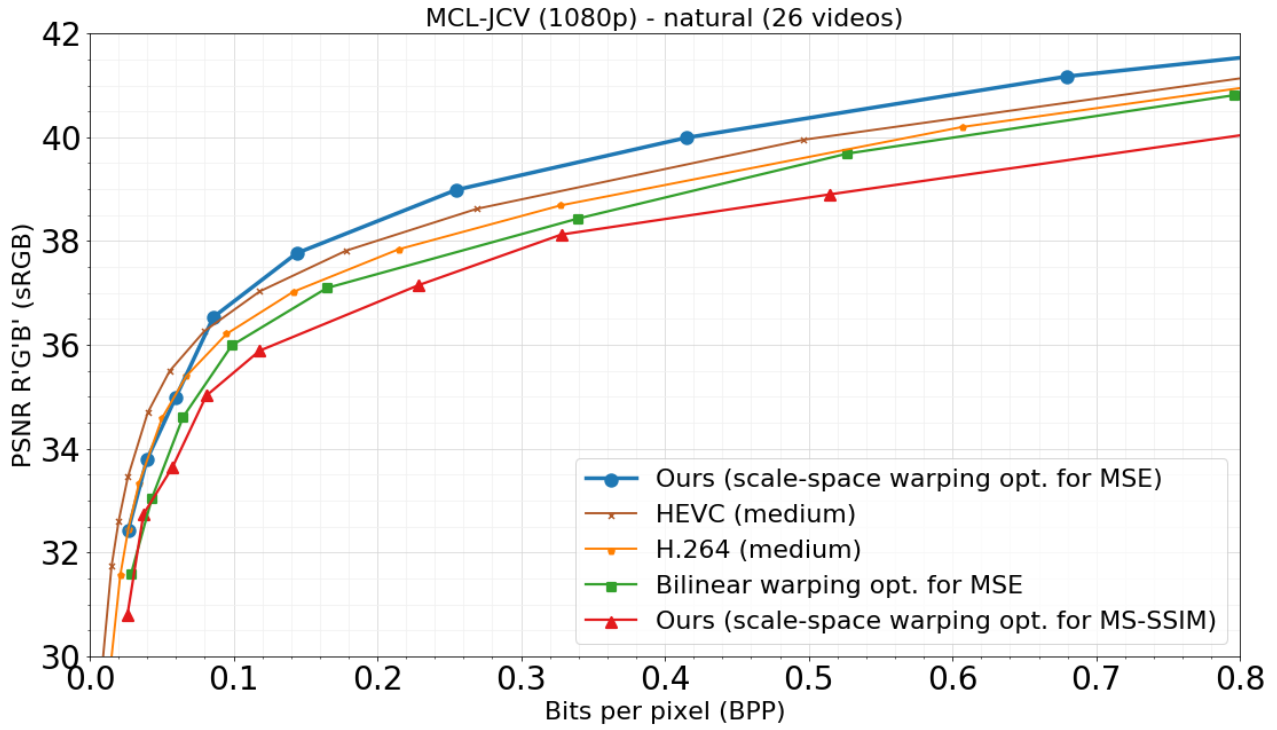
5.2. UVG (MS-SSIM)



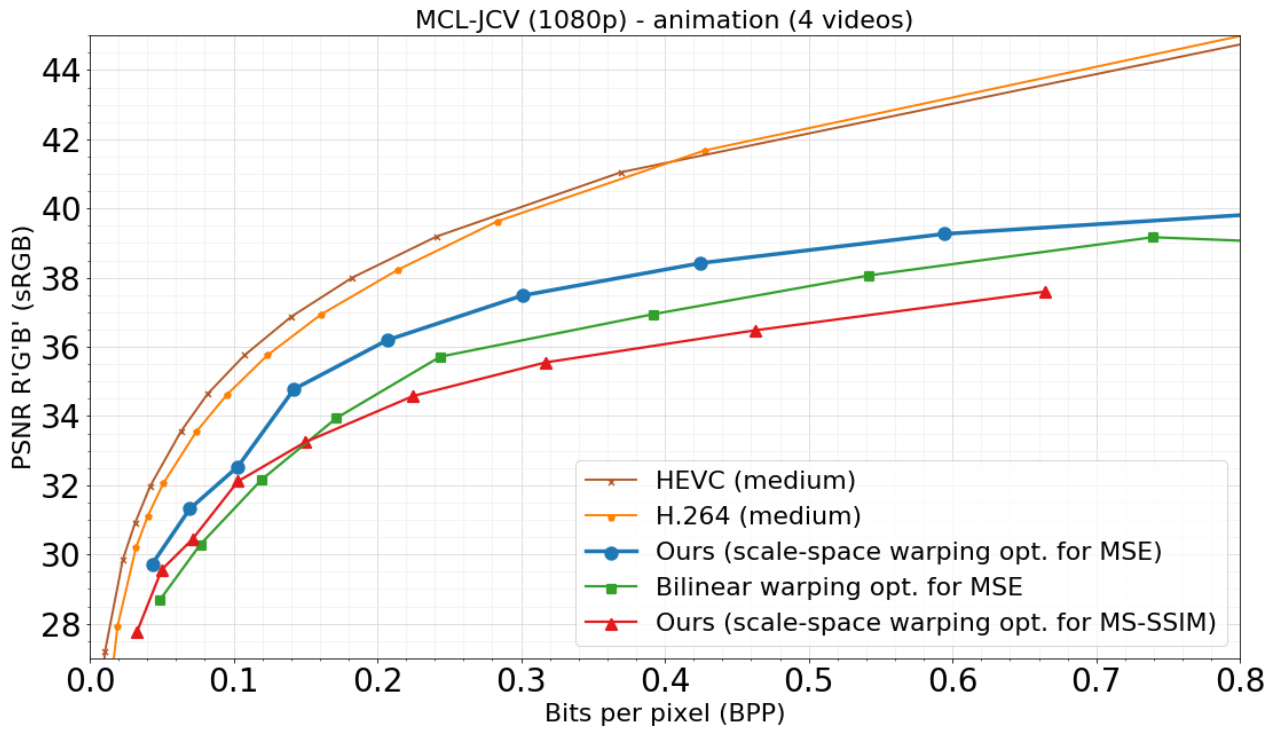
5.3. MCL-JCV (PSNR)



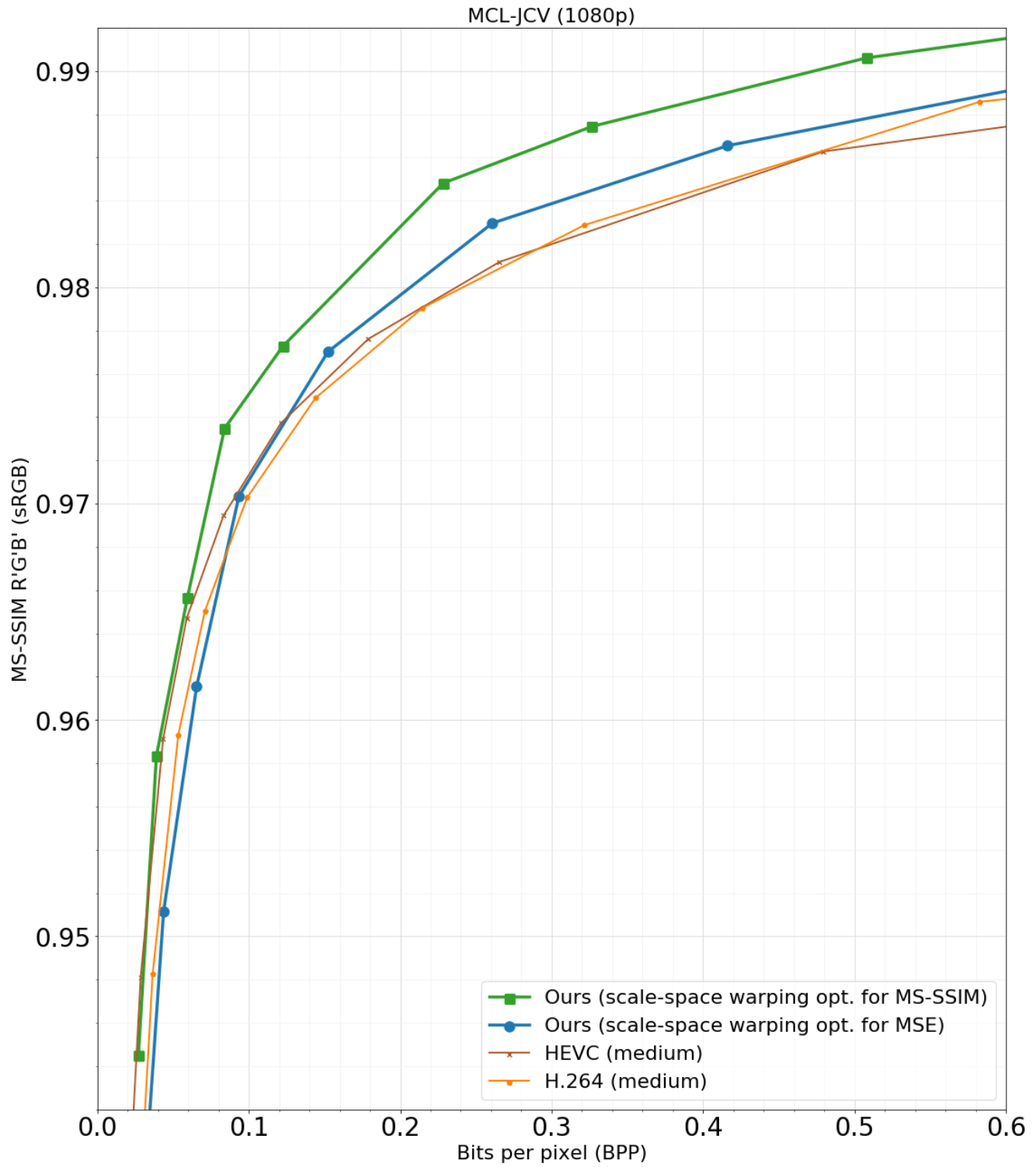
5.3.1 Only "Natural" Videos in MCL-JCV (PSNR)



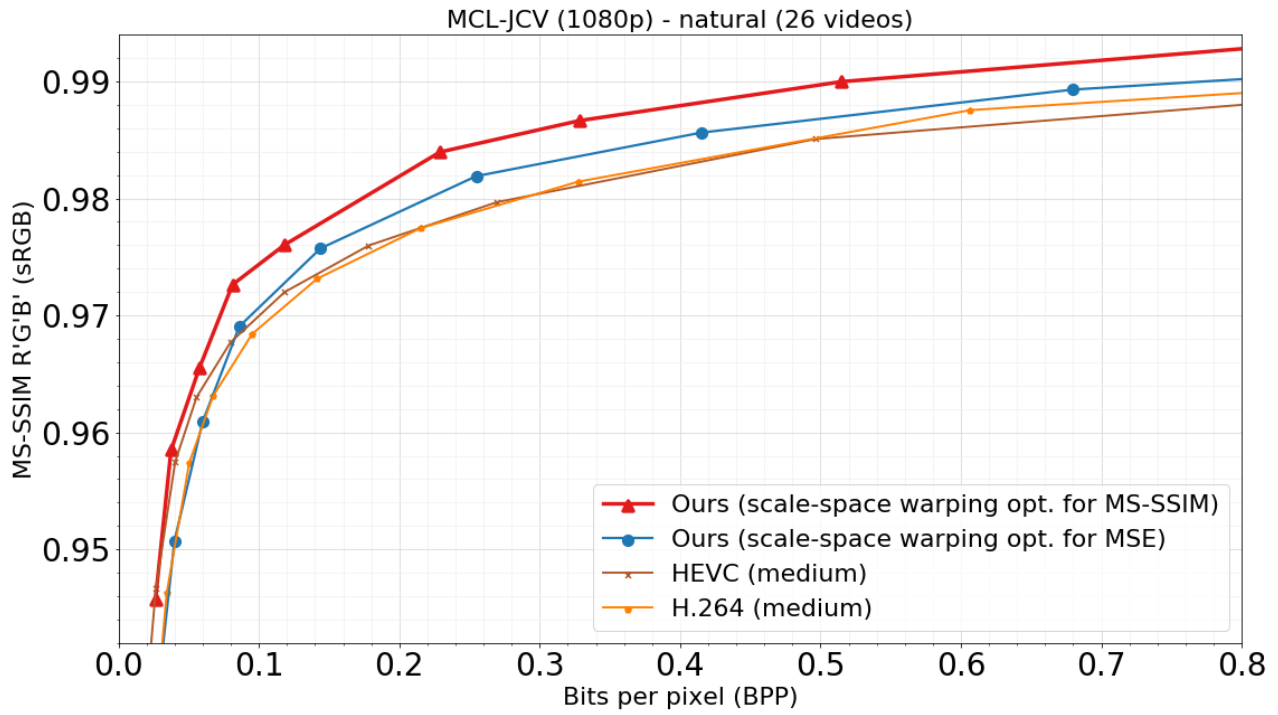
5.3.2 Only Animated Videos in MCL-JCV (PSNR)



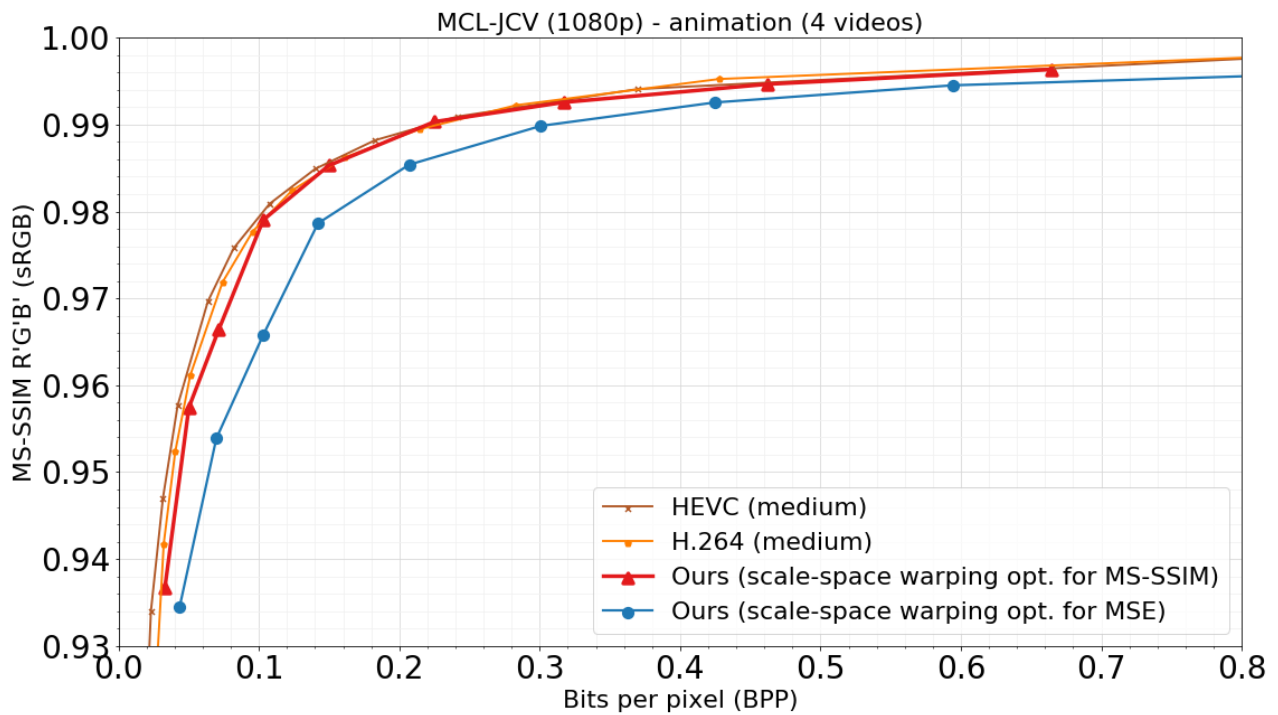
5.4. MCL-JCV (MS-SSIM)



5.4.1 Only "Natural" Videos in MCL-JCV (MS-SSIM)

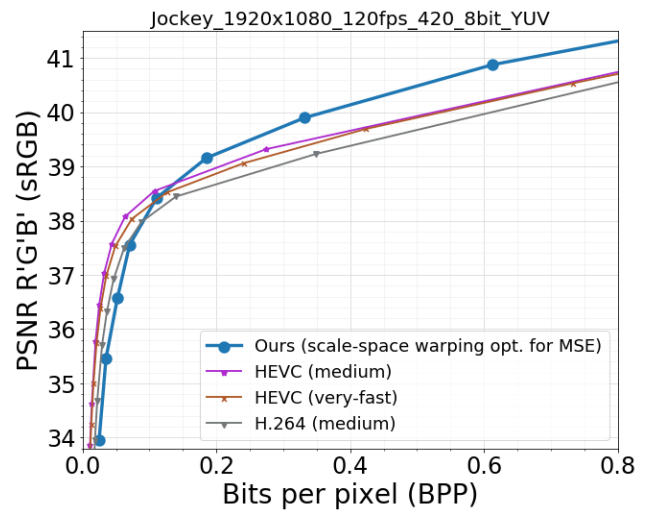
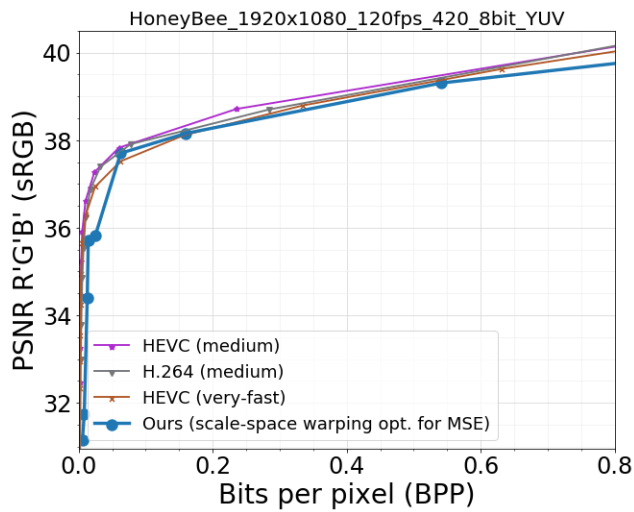
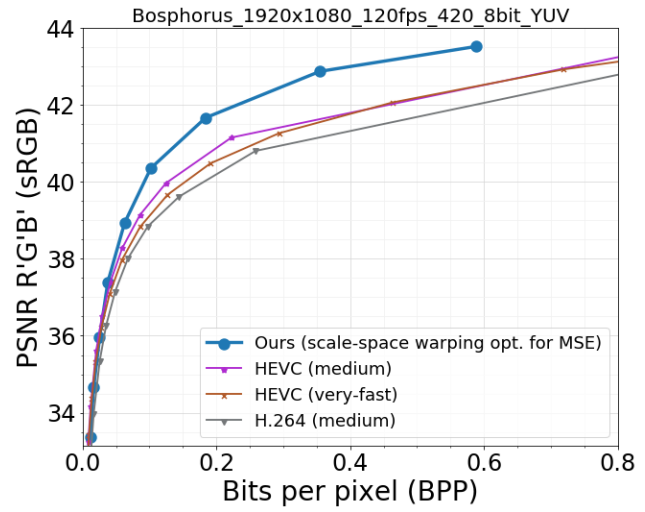
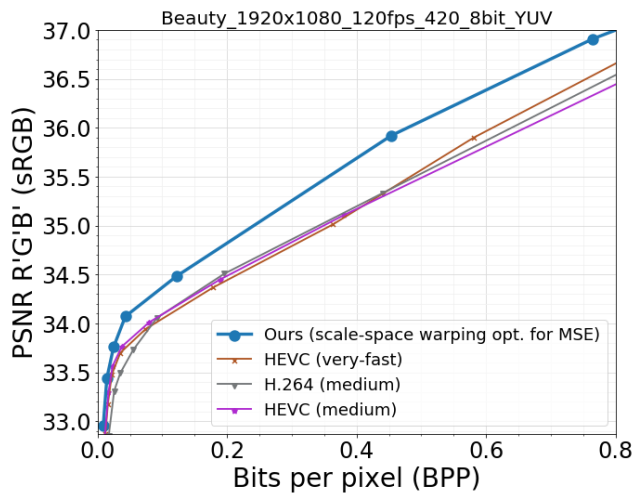


5.4.2 Only Animated Videos in MCL-JCV (MS-SSIM)



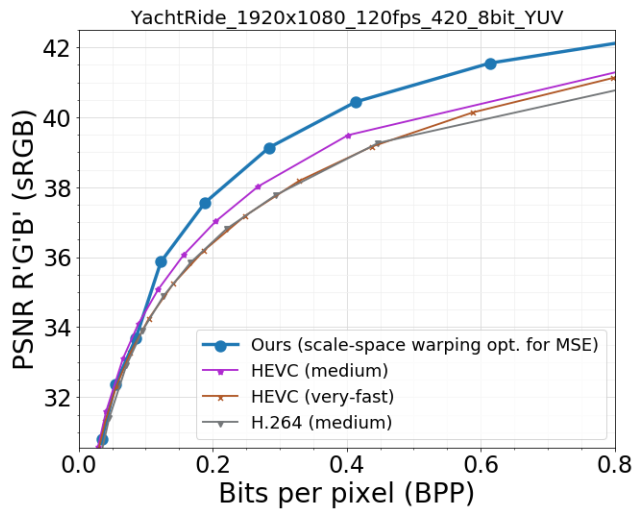
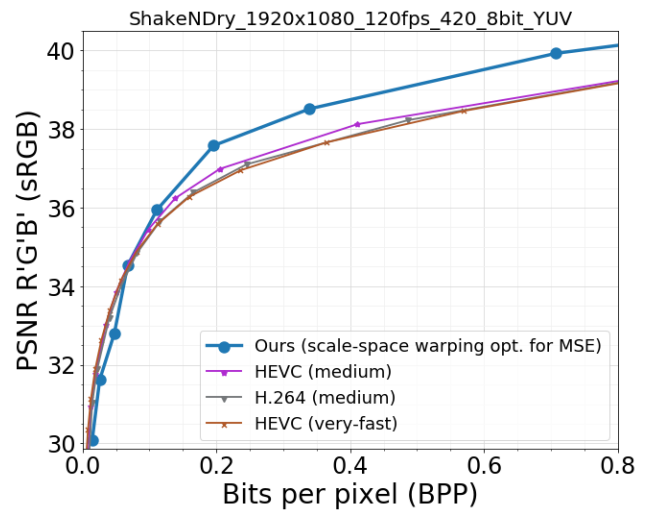
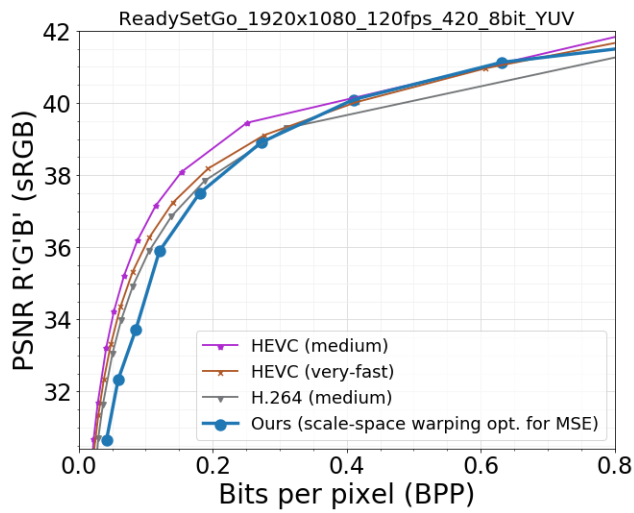
6. Per-Video Rate-Distortion Graphs

6.1. UVG

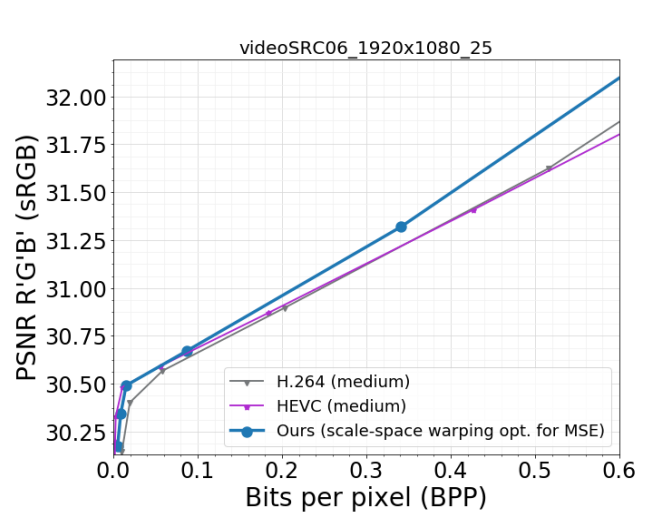
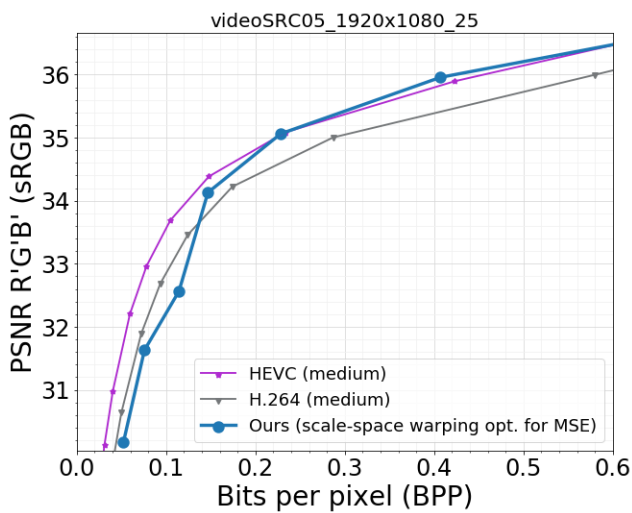
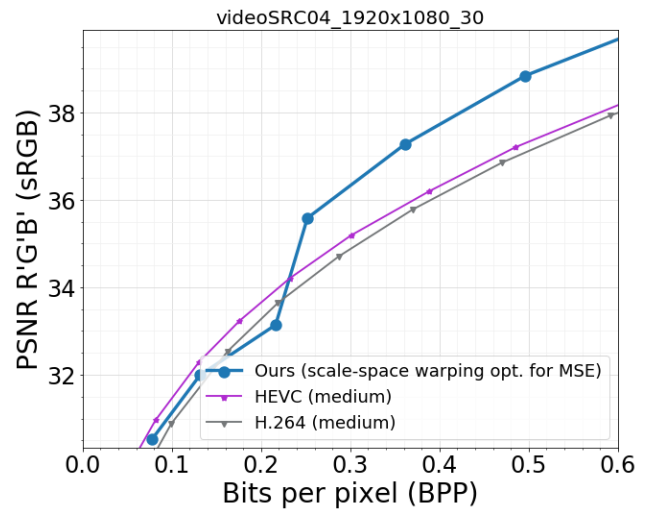
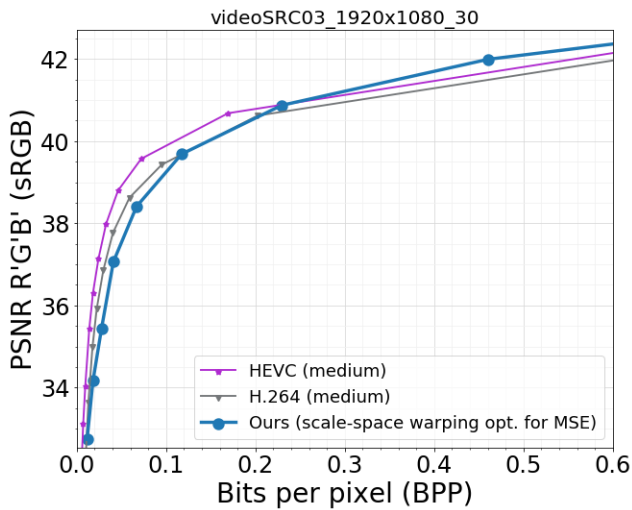
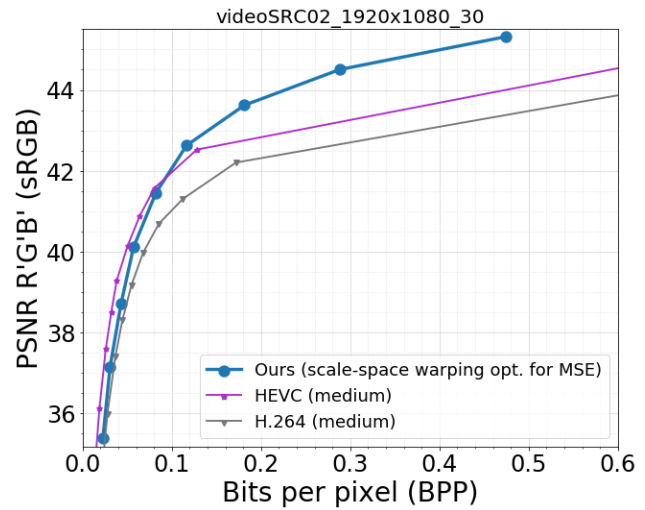
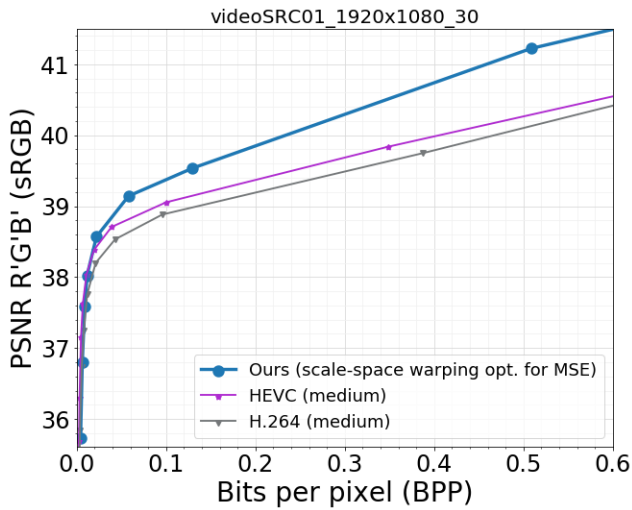


1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

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

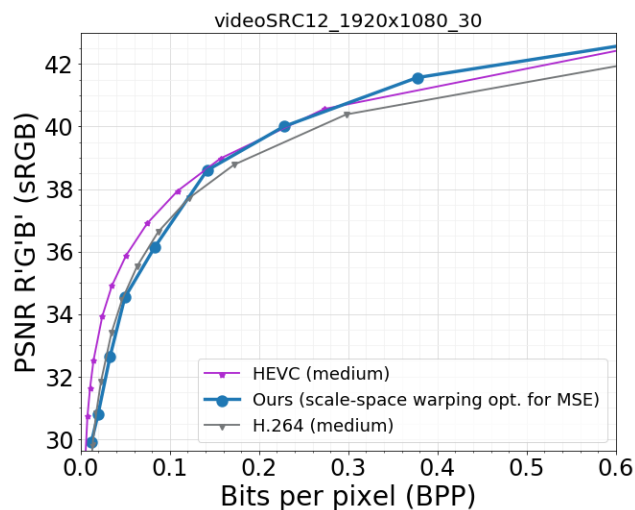
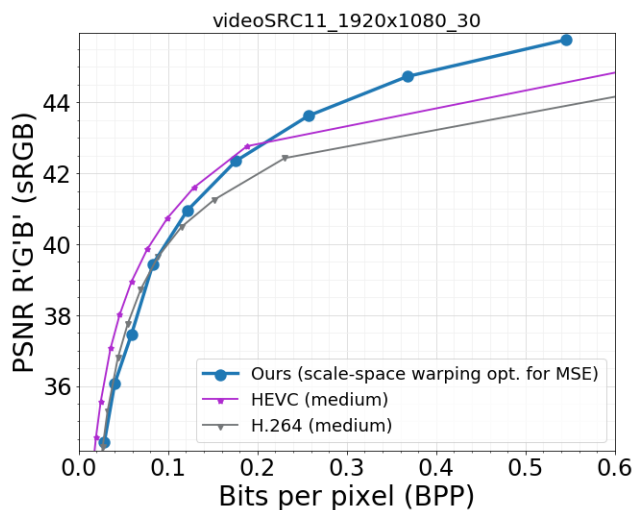
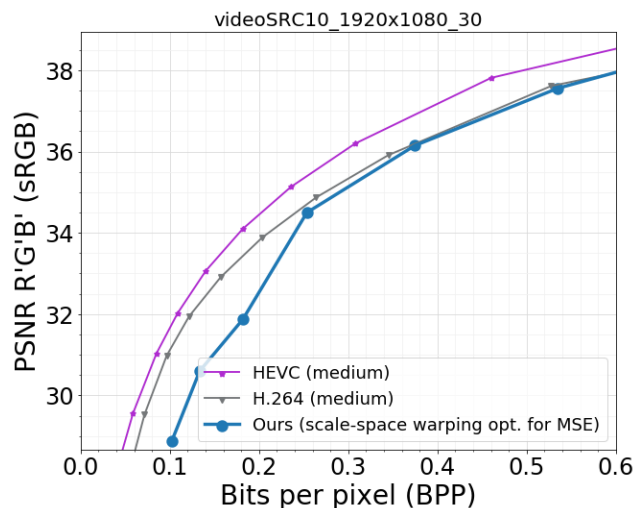
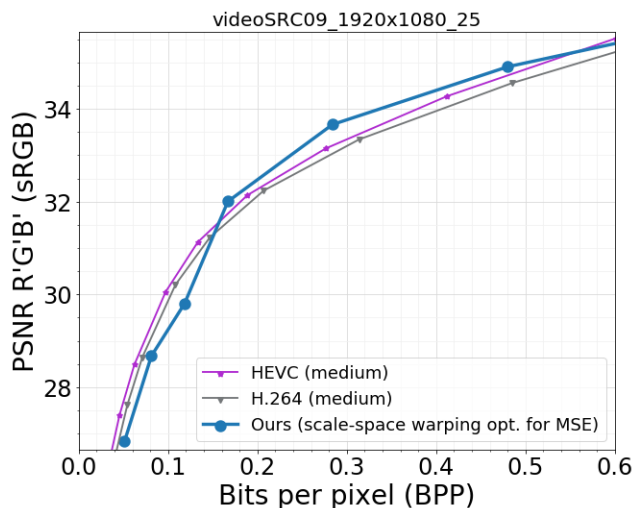
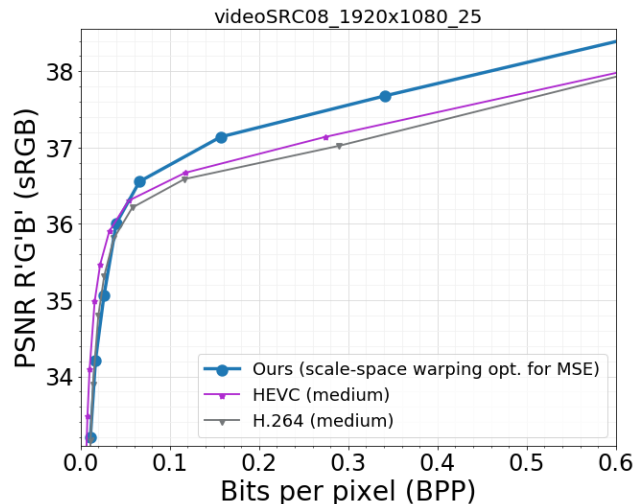
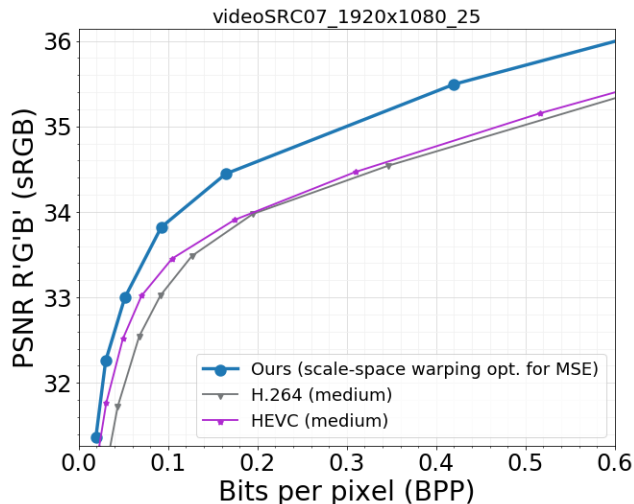


6.2. MCL-JCV

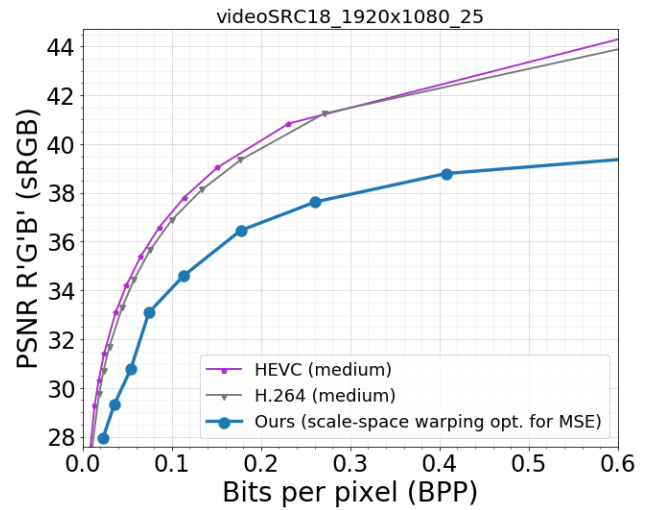
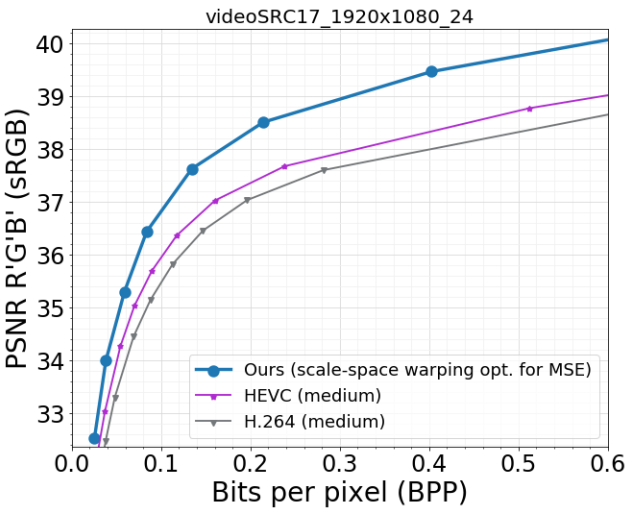
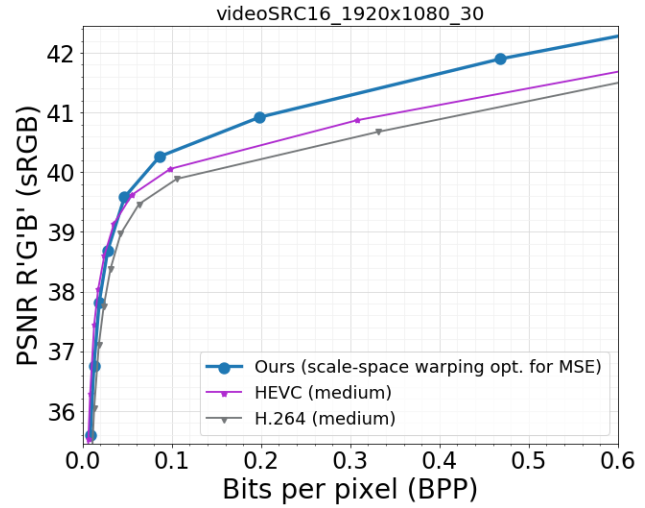
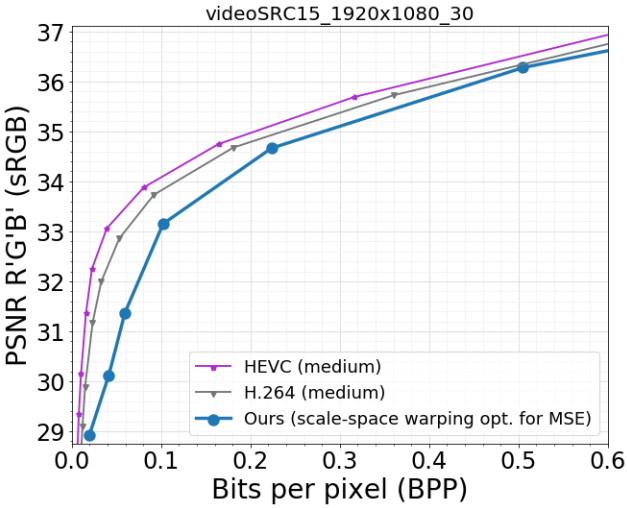
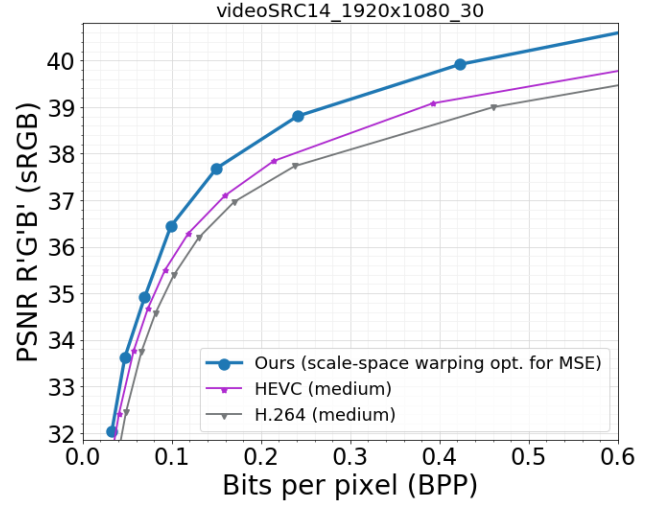
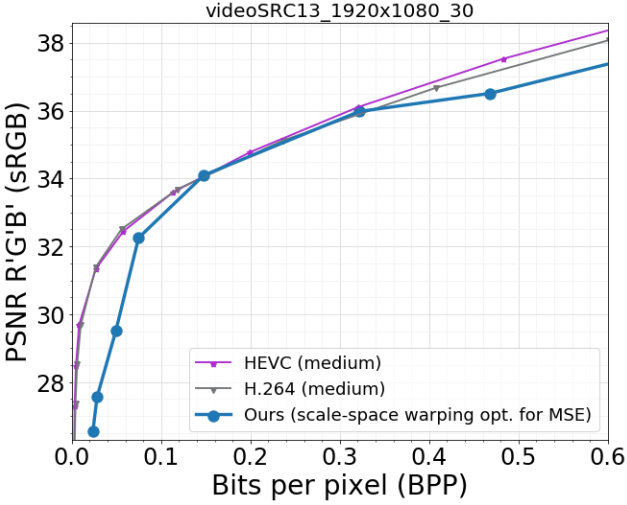


1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565

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



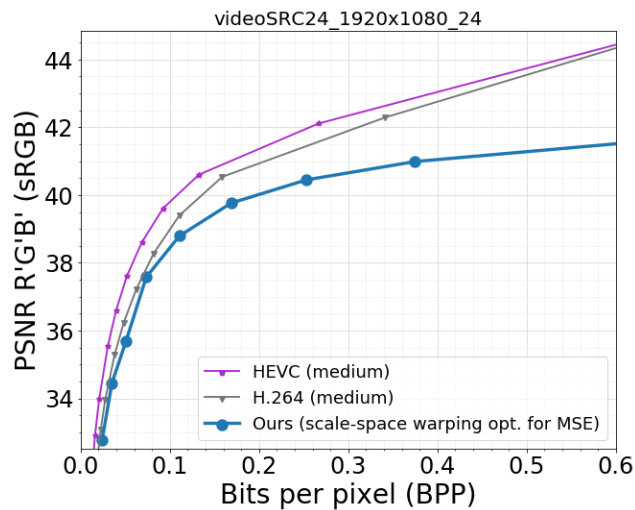
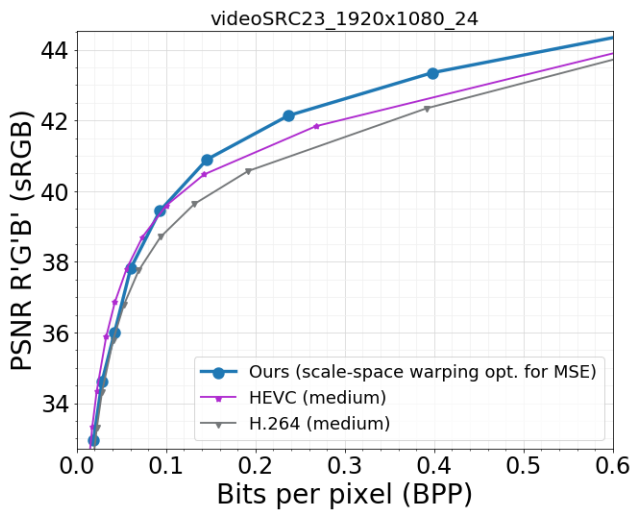
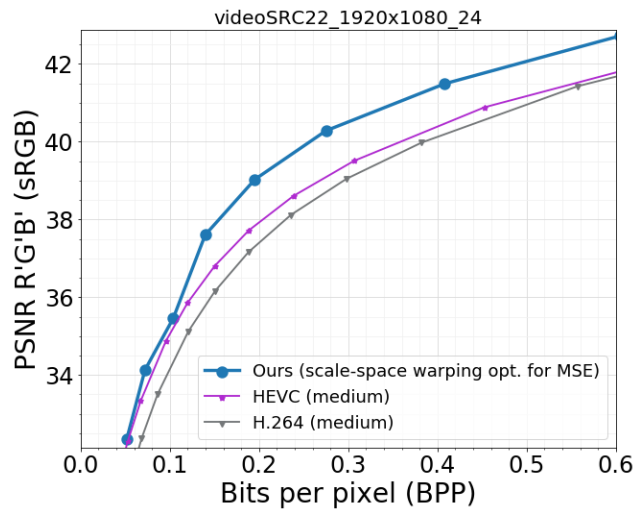
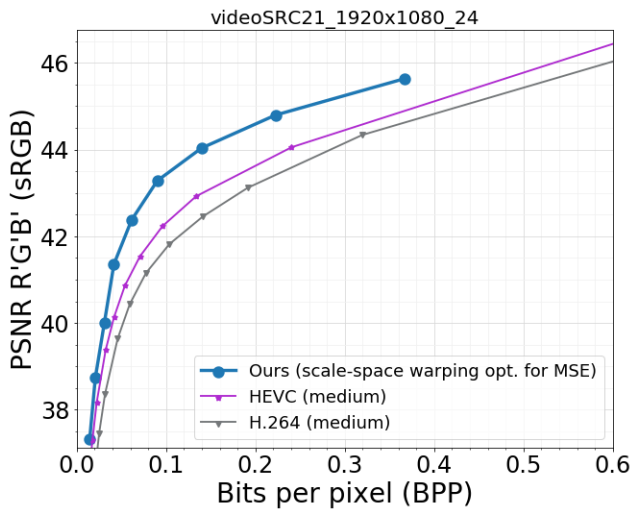
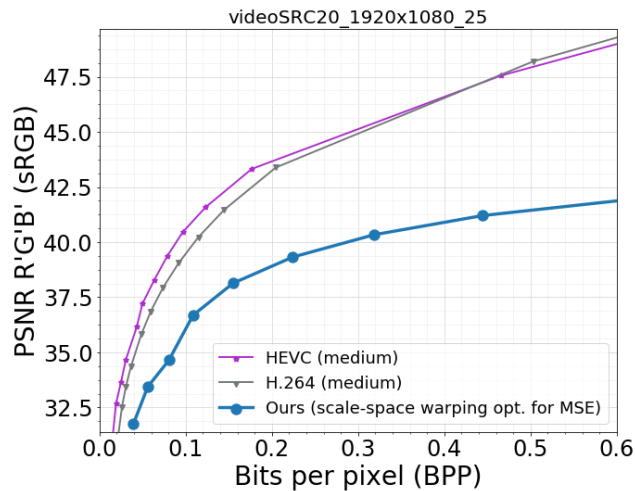
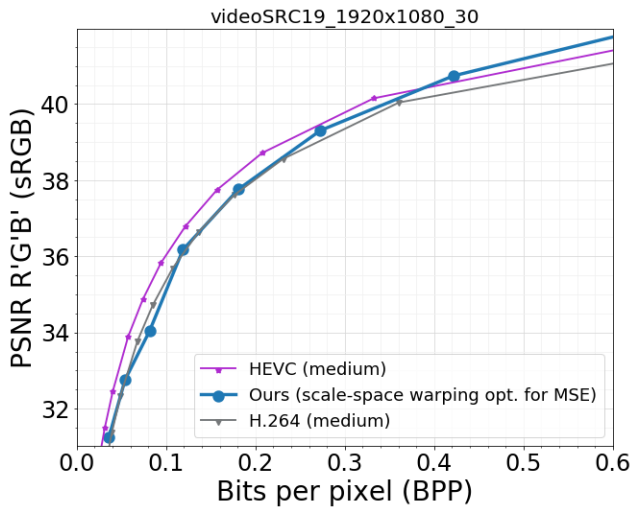
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673



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

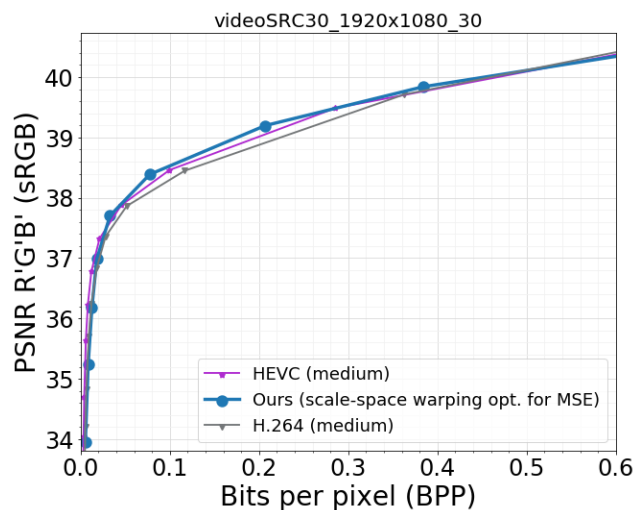
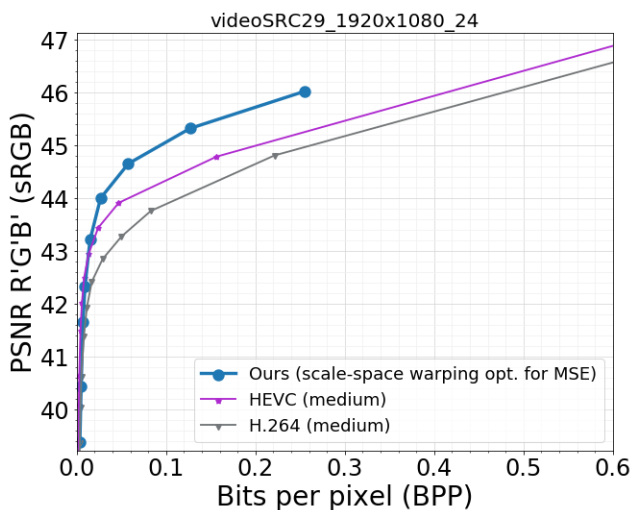
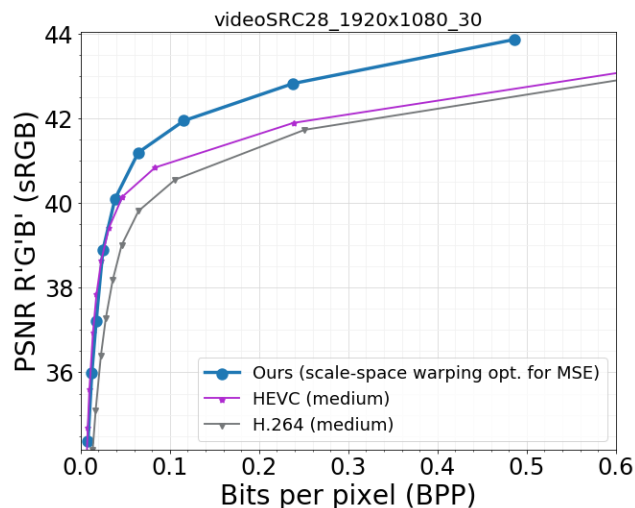
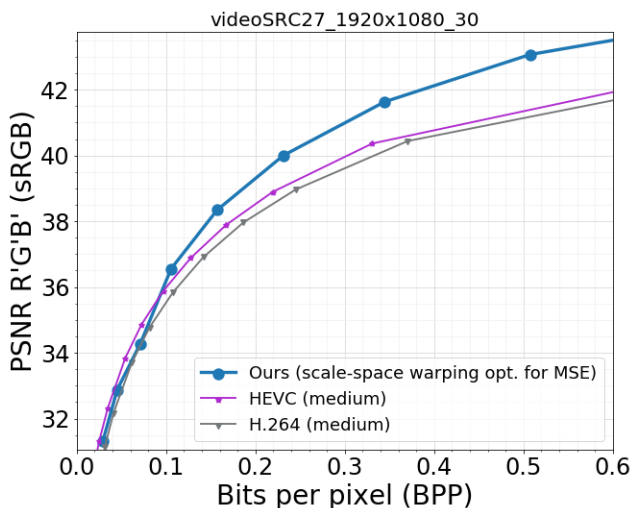
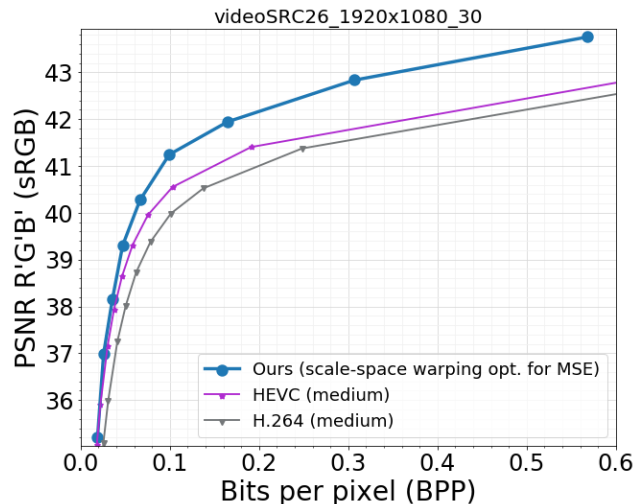
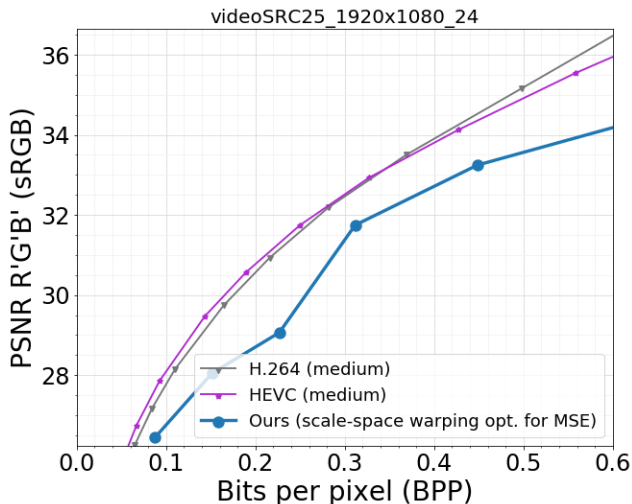
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781

1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835



1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889

1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943



1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997**References**

- [1] FFmpeg. <http://www.ffmpeg.org/>. 1
- [2] Johannes Ballé, Nick Johnston, and David Minnen. Integer networks for data compression with latent-variable models. 2018. 2
- [3] Gisle Bjøntegaard. Calculation of average PSNR differences between RD-curves. Doc. VCEG-M33, ITU-T SG16/Q6 VCEG, Austin, TX, USA, Apr. 2001. 2
- [4] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019. 2

1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051