

000 054
001 055
002 056
003 057
004 058
005 059
006 060
007 061
008 062
009 063
010 064
011 065
012 066
013 067
014 068
015 069
016 070
017 071
018 072
019 073
020 074
021 075
022 076
023 077
024 078
025 079
026 080
027 081
028 082
029 083
030 084
031 085
032 086
033 087
034 088
035 089
036 090
037 091
038 092
039 093
040 094
041 095
042 096
043 097
044 098
045 099
046 100
047 101
048 102
049 103
050 104
051 105
052 106
053 107

Supplementary material for Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network

Anonymous CVPR submission

Paper ID 2823

1. Training and Architecture Details

The training procedure is similar to the one used in standard GANs, which alternatively updates the generator and discriminators until converge.

The Adam optimizer [3] is used. The initial learning rate is set as 0.0002 and decreased by half for every 100 epochs (50 for COCO). The model is trained for 500 epochs in total (200 epochs for COCO). We configure the side outputs at 4 different scales where the feature map resolution is equal to 64^2 , 128^2 , 256^2 , and 512^2 , respectively. For the local image loss of these 4 side outputs, we set $R_1 = 1$, $R_2 = 1$, $R_3 = 5$, and $R_4 = 5$. For example, R_1 refers to 64^2 . These numbers are not fine-tuned but are set empirically. We believe there exists better configurations to be explored.

All intermediate conv layers, except from the specified ones in Section 3.4, use 3×3 kernels (with reflection padding). Some other normalization also layers are experimented (i.e. instance normalization [5] and layer normalization [1]) since they are used by recent advances [7, 2]. But the results are not satisfactory.

With respect to the generator, we use 1-repeat residual blocks for the generator till the 256^2 resolution. The input of the generator is a $1024 \times 4 \times 4$ tensor. As the feature map resolution increases by 2, the number of feature maps is halved at 8, 32, 128, 256 sizes. To generate 512^2 images, we pre-train the generator to 256^2 due to the limitation of the GPU memory. We use a 3-repeat res-block followed by a stretching layer to upscale the feature map size to $32 \times 512 \times 512$. and a linear compression layer to generate images. Since the 256^2 image already captures the overall semantics and details, to boost the training and encourage the 512^2 maintain this information, we use a l1 reconstruction loss to ‘self-regularize’ the generator.

All source code will be released.

2. More Qualitative Results and Analysis

In this section, we demonstrate more sample results for the three datasets.

Figure 1 compares our results with StackGAN. For each input, 6 images are randomly sampled. Furthermore, we visualize zoomed-in samples compared with StackGAN in Figure 2. Our results demonstrate obviously better quality, less artifacts, and less sharp pixel transitions.

Figure 3 shows the results on the CUB bird dataset. All the outputs of a model with different resolutions are also shown. As can be observed in this two figures, our method can generate fairly vivid images with different poses, shape, background, etc. Moreover, the images with different resolutions, which are side outputs of a single model, have very consistent information. More and more image details can be observed as the resolution increases. Figure 4 shows the results on the Oxford-102 flower dataset. Very detailed petals can be generated with photographic colors and saturability.

Figure 5 shows some sampled results on the COCO dataset. COCO is much more challenging than the other two datasets since it contains natural images from a wide variety of scenes containing hundreds of different objects. As can be observed in the shown samples, our method can still generate semantically consistent images.

However, it is worth to notice that, although our method significantly improves existing methods [6, 4] on COCO, we realize that generating fine-grained details of complex natural scenes with various objects is still challenging. Based on this study, we expect to further address this problem as the future study.

108
109
110
111
112

This is a tiny bird with a large protruding tan chest and a short black beak that has grey wings

113
114
115
116
117
118
119
120
121
122
123162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178124
125
126
127
128179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194129
130
131
132
133
134
135
136
137
138
139
140

Medium sized bird, red crown to orange fades to his tail, organge abdomen and belly, wing edges are gray

195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211141
142
143
144
145212
213
214
215146
147
148
149
150
151
152
153
154
155
156

This small bird has black wings and a yellow and black spotted belly along with a small, pointy beak

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
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
990
991
992
993
994
995
996
997
998
999
1000

Figure 1. Sample results on CUB compared with StackGAN. For each input, 6 samples are shown with resolutions of 64^2 and 256^2 , respectively. As can be obviously seen, our HDGAN can generate very consistent content in images of different resolutions. Moreover, our generated images show more photographic color and contrast.

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

Zoom-in

Zoom-in

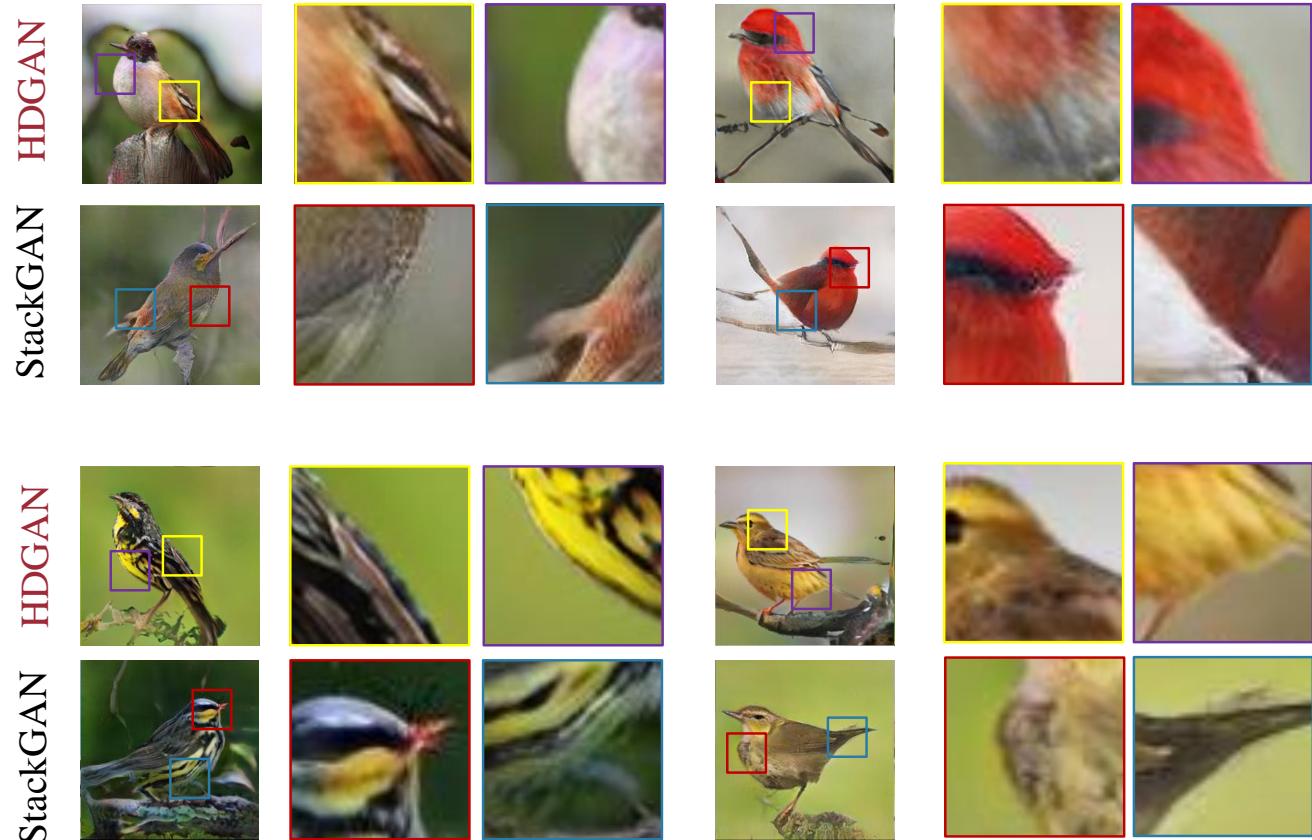
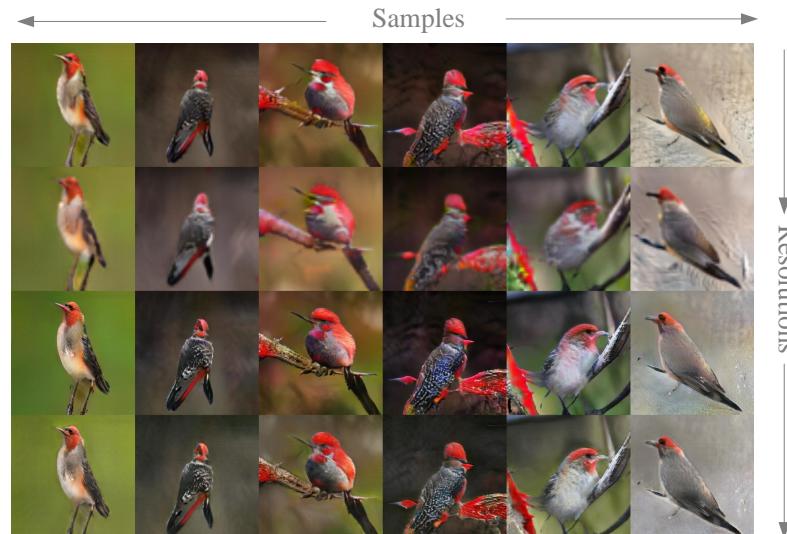


Figure 2. Zoomed-in samples compared with StackGAN. The best sample among 6 samples given an input text is selected. It can be clearly observed that our results show more smooth visual results. Especially, much less sharp pixel transitions exist in our results.

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 A small bird with a red
333 crown and straight bill
334 sits perched atop a
335 branch
336
337
338
339
340
341



342
343
344
345
346
347
348
349
350 A small yellow/green
351 bird with black wings
352 and white wingbars
353
354
355
356
357



358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374 The bird has sharp pointed
375 beak with grayish yellow
376 throat, with white eyering
377

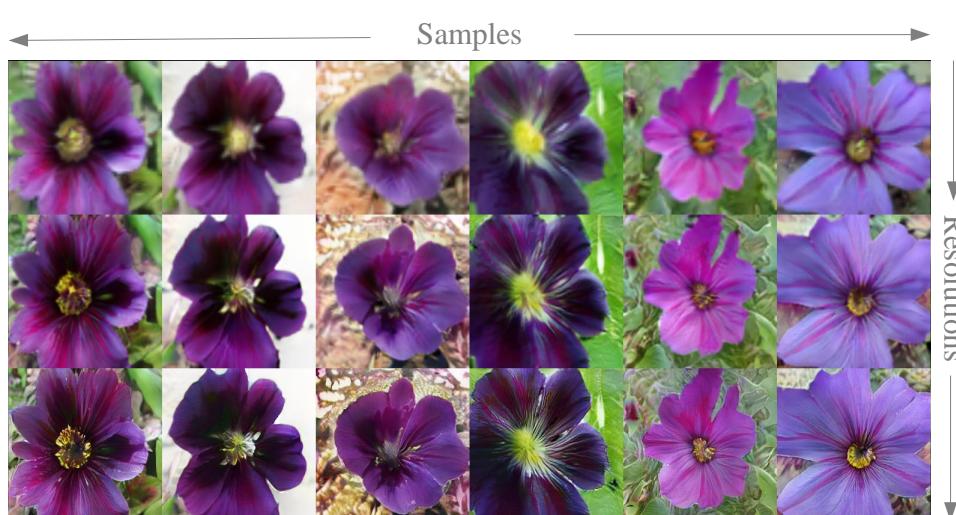


Figure 3. Sample results on CUB. For each input, 6 samples with resolutions of 64^2 , 128^2 , 256^2 , and 512^2 are shown in 4 rows, respectively.

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

This flower is purple and red in color, and has petals that are striped

449
450
451
452
453
454
455
456
457
458

This flower is pink and yellow in color, with petals that are skinny and oval.

464
465
466
467
468
469

Flower has petals that are pale pink with yellow stamen

479
480

Figure 4. Sample results on Oxford-102. For each input, 6 samples with resolutions of 64^2 , 128^2 , and 256^2 are shown in 3 rows, respectively.

481

References

484
485

- [1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [2] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *ICCV*, 2017.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502503
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
532533
534
535
536
537
538
539

540	A close up of a plate of food containing broccoli		594
541	A photo of a train on a bridge above a river		595
542	A beach on a sunny day with a bunch of people on it.		596
543	A living room filled with lots of furniture		597
544			598
545			599
546			600
547			601
548			602
549			603
550			604
551	A man riding a kiteboard on top of the ocean		605
552	A very tall cathedral towering over a city		606
553	A group of people carrying ski equipment while walking on snow covered ground.		607
554	People playing with kites outside in the desert		608
555			609
556			610
557			611
558			612
559			613
560			614
561	Figure 5. Sample results on COCO. We show 8 256^2 samples in very different scenes.		615
562			616
563	[3] D. Kingma and J. Ba. Adam: A method for stochastic optimization. <i>arXiv preprint arXiv:1412.6980</i> , 2014.		617
564	[4] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. <i>ICML</i> , 2016.		618
565	[5] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. <i>arXiv preprint arXiv:1607.08022</i> , 2016.		619
566	[6] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In <i>ICCV</i> , 2017.		620
567	[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. <i>ICCV</i> , 2017.		621
568			622
569			623
570			624
571			625
572			626
573			627
574			628
575			629
576			630
577			631
578			632
579			633
580			634
581			635
582			636
583			637
584			638
585			639
586			640
587			641
588			642
589			643
590			644
591			645
592			646
593			647