

---

# 000 001 002 003 004 005 006 007 008 009 Improved Denoising Diffusion Probabilistic Models (Supplementary)

---

010  
011 Anonymous Authors<sup>1</sup>

## 012 1. Hyperparameters

013 For all of our experiments, we use a UNet model architecture<sup>1</sup> similar to that used by Ho et al. (2020). We changed  
014 the attention layers to use multi-head attention (Vaswani  
015 et al., 2017), and opted to use four attention heads rather  
016 than one (while keeping the same total number of channels).  
017 We employ attention not only at the 16x16 resolution, but  
018 also at the 8x8 resolution. Additionally, we changed the  
019 way the model conditions on  $t$ . In particular, instead of com-  
020 puting a conditioning vector  $v$  and injecting it into hidden  
021 state  $h$  as GroupNorm( $h + v$ ), we compute conditioning  
022 vectors  $w$  and  $b$  and inject them into the hidden state as  
023 GroupNorm( $h(w + 1) + b$ ). We found in preliminary ex-  
024 periments on ImageNet 64 × 64 that these modifications  
025 slightly improved FID.

026 For ImageNet 64 × 64 the architecture we use is described  
027 as follows. The downsampling stack performs four steps of  
028 downsampling, each with three residual blocks (He et al.,  
029 2015). The upsampling stack is setup as a mirror image of  
030 the downsampling stack. From highest to lowest resolution,  
031 the UNet stages use  $[C, 2C, 3C, 4C]$  channels, respectively.  
032 In our ImageNet 64 × 64 ablations, we set  $C = 128$ , but  
033 we experiment with scaling  $C$  in a later section. We esti-  
034 mate that, with  $C = 128$ , our model is comprised of 120M  
035 parameters and requires roughly 39 billion FLOPs in the  
036 forward pass.

037 For our CIFAR-10 experiments, we use a smaller model with  
038 three resblocks per downsampling stage and layer widths  
039  $[C, 2C, 2C, 2C]$  with  $C = 128$ . We swept over dropout  
040 values  $\{0.1, 0.2, 0.3\}$  and found that 0.1 worked best for  
041 the linear schedule while 0.3 worked best for our cosine  
042 schedule. We expand upon this in Section 6.

043 We use Adam (Kingma & Ba, 2014) for all of our exper-  
044 iments. For most experiments, we use a batch size of 128,

045  
046 <sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region,  
047 Anonymous Country. Correspondence to: Anonymous Author  
048 <anon.email@domain.com>.

049 Preliminary work. Under review by the International Conference  
050 on Machine Learning (ICML). Do not distribute.

051 <sup>1</sup>In initial experiments, we found that a ResNet-style architec-  
052 ture with no downsampling achieved better log-likelihoods but  
053 worse FIDs than the UNet architecture.

$\lambda$	Stop-gradient	NLL	FID
0.001	yes	3.62	28.04
0.01	yes	3.62	27.36
0.001	no	3.62	30.89
0.01	no	3.63	34.35

Table 1. Ablating hyper-parameters of the  $L_{\text{hybrid}}$  objective on ImageNet 64 × 64. All models were trained for 200K iterations.

a learning rate of  $10^{-4}$ , and an exponential moving average (EMA) over model parameters with a rate of 0.9999. For our scaling experiments, we vary the learning rate to accomodate for different model sizes. For our larger class-conditional ImageNet 64 × 64 experiments, we scaled up the batch size to 2048 for faster training on more GPUs.

When using the linear noise schedule from Ho et al. (2020), we linearly interpolate from  $\beta_1 = 0.0001/4$  to  $\beta_{4000} = 0.02/4$  to preserve the shape of  $\bar{\alpha}_t$  for the  $T = 4000$  schedule.

When computing FID we produce 50K samples from our models, except for unconditional ImageNet 64 × 64 where we produce 10K samples. Using only 10K samples biases the FID to be higher, but requires much less compute for sampling and helps do large ablations. Since we mainly use FID for relative comparisons on unconditional ImageNet 64 × 64, this bias is acceptable. For computing the reference distribution statistics we follow prior work (Ho et al., 2020; Brock et al., 2018) and use the full training set for CIFAR-10 and ImageNet, and 50K training samples for LSUN. Note that unconditional ImageNet 64 × 64 models are trained and evaluated using the official ImageNet-64 dataset (van den Oord et al., 2016), whereas for class conditional ImageNet 64 × 64 and 256 × 256 we center crop and area downsample images (Brock et al., 2018).

In Table 1 we ablate the two major choices in our  $L_{\text{hybrid}}$  objective: the  $L_{\text{vib}}$  weight  $\lambda$ , and the stop-gradient after  $\mu_\theta$  when computing  $L_{\text{vib}}$ . We find that the stop-gradient improves sample quality and reduces sensitivity to  $\lambda$ .

## 2. Fast Sampling on LSUN 256 × 256

To test the effectiveness of our  $L_{\text{hybrid}}$  models on a high-resolution domain, we trained both  $L_{\text{hybrid}}$  and  $L_{\text{simple}}$  models on the LSUN bedroom (Yu et al., 2015) dataset. We

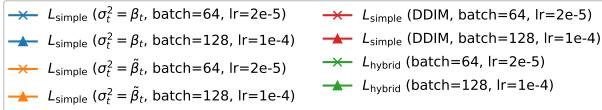


Figure 1. FID vs. number of sampling steps from an LSUN 256 × 256 bedroom model.

MODEL	FID
VQ-VAE-2 ((Razavi et al., 2019), two-stage)	38.1
Improved Diffusion (ours, single-stage)	31.5
Improved Diffusion (ours, two-stage)	<b>12.3</b>
BigGAN (Brock et al., 2018)	7.7
BigGAN-deep (Brock et al., 2018)	<b>7.0</b>

Table 2. Sample quality comparison on class conditional ImageNet 256 × 256. BigGAN FIDs are reported for the truncation that results in the best FID.

train two models: one with batch size 64 and learning rate  $2 \times 10^{-5}$  as in Ho et al. (2020), and another with a larger batch size 128 and learning rate  $10^{-4}$ . All models were trained with 153.6M examples, which is 2.4M training iterations with batch size 64.

Our results are displayed in Figure 1. We find that DDIM outperforms our  $L_{\text{hybrid}}$  model when using fewer than 50 diffusion steps, while our  $L_{\text{hybrid}}$  model outperforms DDIM with more than 50 diffusion steps. Interestingly, we note that DDIM benefits from a smaller learning rate and batch size, whereas our method is able to take advantage of a larger learning rate and batch size.

### 3. Sample Quality on ImageNet 256 × 256

We trained two models on class conditional ImageNet 256 × 256. The first is a usual diffusion model that directly models the 256 × 256 images. The second model reduces compute by chaining a pretrained 64 × 64 model  $p(x_{64}|y)$  with another upsampling diffusion model  $p(x_{256}|x_{64}, y)$  to upsample images to 256 × 256. For the upsampling model, the downsampled image  $x_{64}$  is passed as extra conditioning

input to the UNet. This is similar to VQ-VAE-2 (Razavi et al., 2019), which uses two stages of priors at different latent resolutions to more efficiently learn global and local features. The linear schedule worked better for 256 × 256 images, so we used that for these results. Table 2 summarizes our results. For VQ-VAE-2, we use the FIDs reported in (Ravuri & Vinyals, 2019). Diffusion models still obtain the best FIDs for a likelihood-based model, and close the gap to GANs considerably.

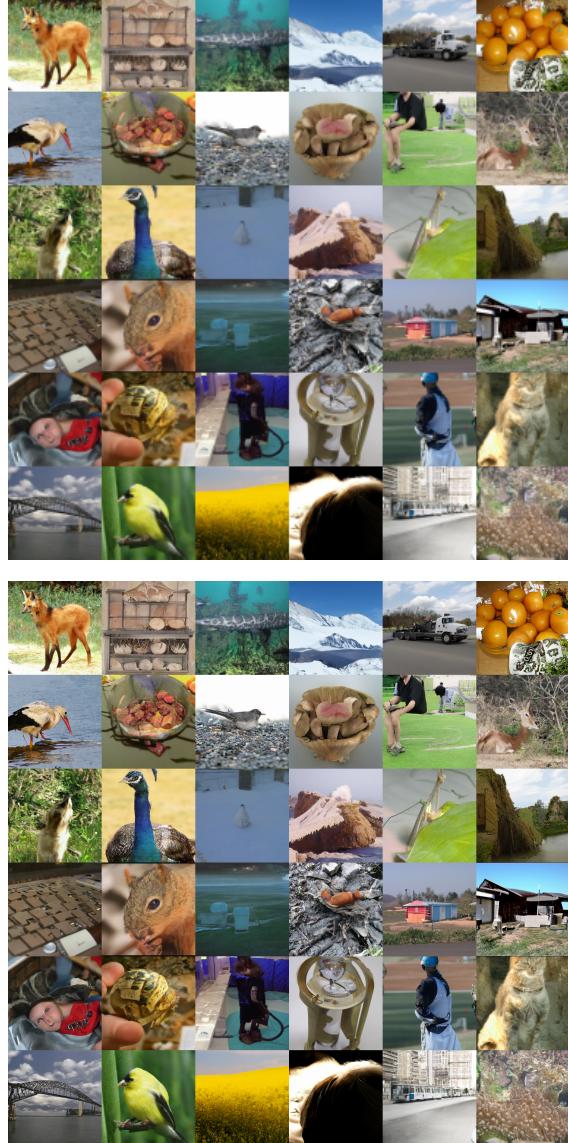
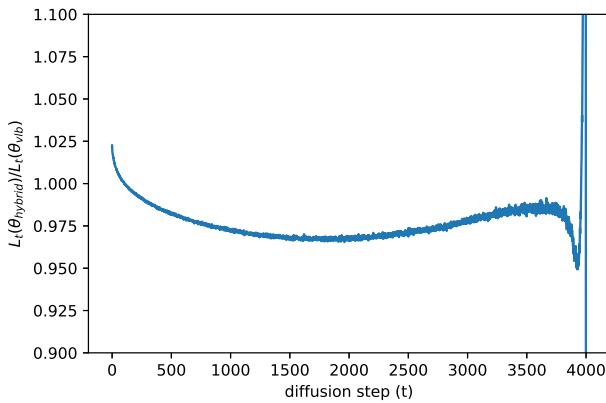
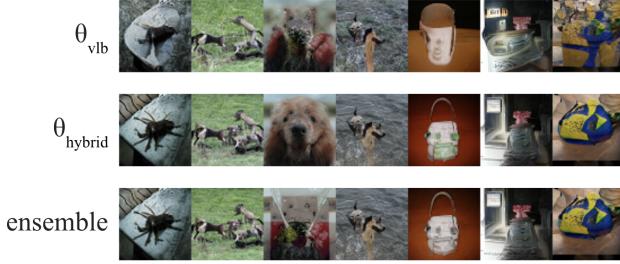


Figure 2. Random samples from two-stage class conditional ImageNet 256 × 256 model. On top are random samples from the 64 × 64 model (FID 2.92), whereas on bottom are the results after upsampling them to 256 × 256 (FID 12.3). Each model uses 250 sampling steps.

110  
 111  
 112  
**4. Combining  $L_{\text{hybrid}}$  and  $L_{\text{vlb}}$  Models**  
 113  
 114  
 115  


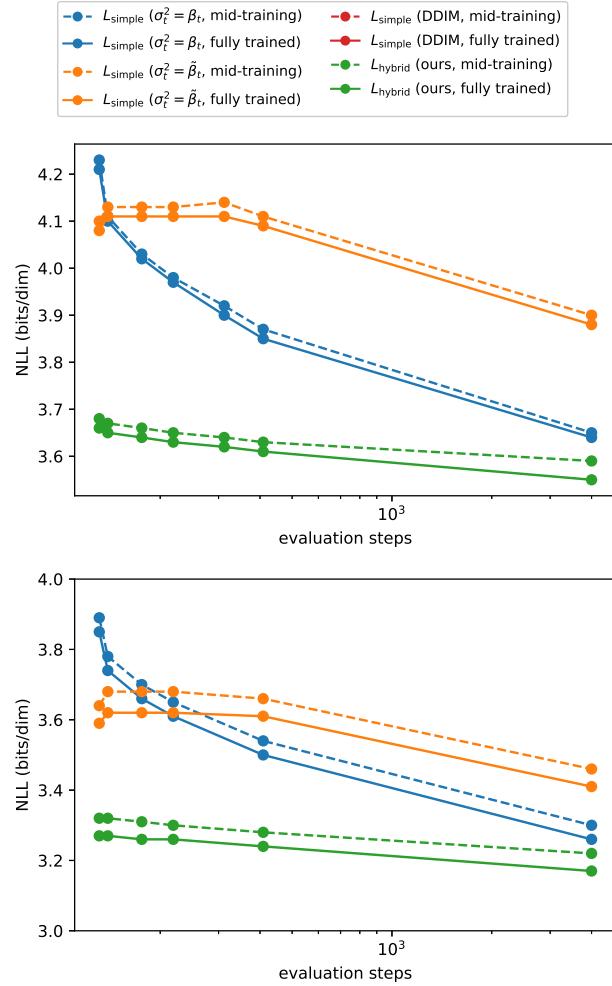
126  
 127  
 128  
 Figure 3. The ratio between VLB terms for each diffusion step of  
 129  
 $\theta_{\text{hybrid}}$  and  $\theta_{\text{vlb}}$ . Values less than 1.0 indicate that  $\theta_{\text{hybrid}}$  is "better"  
 130  
 than  $\theta_{\text{vlb}}$  for that timestep of the diffusion process.  
 131  
 132



141  
 142  
 143  
 144  
 Figure 4. Samples from  $\theta_{\text{vlb}}$  and  $\theta_{\text{hybrid}}$ , as well as an ensemble  
 145  
 produced by using  $\theta_{\text{vlb}}$  for the first and last 100 diffusion steps. For  
 146  
 these samples, the seed was fixed, allowing a direct comparison  
 147  
 between models.  
 148  
 149  
 150  
 151  
 152

To understand the trade-off between  $L_{\text{hybrid}}$  and  $L_{\text{vlb}}$ , we show in Figure 3 that the model resulting from  $L_{\text{vlb}}$  (referred to as  $\theta_{\text{vlb}}$ ) is better at the start and end of the diffusion process, while the model resulting from  $L_{\text{hybrid}}$  (referred to as  $\theta_{\text{hybrid}}$ ) is better throughout the middle of the diffusion process. This suggests that  $\theta_{\text{vlb}}$  is focusing more on imperceptible details, hence the lower sample quality.

Given the above observation, we performed an experiment on ImageNet  $64 \times 64$  to combine the two models by constructing an ensemble that uses  $\theta_{\text{hybrid}}$  for  $t \in [100, T - 100]$  and  $\theta_{\text{vlb}}$  elsewhere. We found that this model achieved an FID of **19.9** and an NLL of **3.52 bits/dim**. This is only slightly worse than  $\theta_{\text{hybrid}}$  in terms of FID, while being better than both models in terms of NLL.

 160  
 161  
 162  
**5. Log-likelihood with Fewer Diffusion Steps**  
 163  
 164


165  
 166  
 167  
 168  
 Figure 5. NLL versus number of evaluation steps, for models  
 169  
 trained on ImageNet  $64 \times 64$  (top) and CIFAR-10 (bottom). All  
 170  
 models were trained with 4000 diffusion steps.  
 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  
 560  
 561  
 562  
 563  
 564  
 565  
 566  
 567  
 568  
 569  
 570  
 571  
 572  
 573  
 574  
 575  
 576  
 577  
 578  
 579  
 580  
 581  
 582  
 583  
 584  
 585  
 586  
 587  
 588  
 589  
 590  
 591  
 592  
 593  
 594  
 595  
 596  
 597  
 598  
 599  
 600  
 601  
 602  
 603  
 604  
 605  
 606  
 607  
 608  
 609  
 610  
 611  
 612  
 613  
 614  
 615  
 616  
 617  
 618  
 619  
 620  
 621  
 622  
 623  
 624  
 625  
 626  
 627  
 628  
 629  
 630  
 631  
 632  
 633  
 634  
 635  
 636  
 637  
 638  
 639  
 640  
 641  
 642  
 643  
 644  
 645  
 646  
 647  
 648  
 649  
 650  
 651  
 652  
 653  
 654  
 655  
 656  
 657  
 658  
 659  
 660  
 661  
 662  
 663  
 664  
 665  
 666  
 667  
 668  
 669  
 670  
 671  
 672  
 673  
 674  
 675  
 676  
 677  
 678  
 679  
 680  
 681  
 682  
 683  
 684  
 685  
 686  
 687  
 688  
 689  
 690  
 691  
 692  
 693  
 694  
 695  
 696  
 697  
 698  
 699  
 700  
 701  
 702  
 703  
 704  
 705  
 706  
 707  
 708  
 709  
 710  
 711  
 712  
 713  
 714  
 715  
 716  
 717  
 718  
 719  
 720  
 721  
 722  
 723  
 724  
 725  
 726  
 727  
 728  
 729  
 730  
 731  
 732  
 733  
 734  
 735  
 736  
 737  
 738  
 739  
 740  
 741  
 742  
 743  
 744  
 745  
 746  
 747  
 748  
 749  
 750  
 751  
 752  
 753  
 754  
 755  
 756  
 757  
 758  
 759  
 760  
 761  
 762  
 763  
 764  
 765  
 766  
 767  
 768  
 769  
 770  
 771  
 772  
 773  
 774  
 775  
 776  
 777  
 778  
 779  
 780  
 781  
 782  
 783  
 784  
 785  
 786  
 787  
 788  
 789  
 790  
 791  
 792  
 793  
 794  
 795  
 796  
 797  
 798  
 799  
 800  
 801  
 802  
 803  
 804  
 805  
 806  
 807  
 808  
 809  
 810  
 811  
 812  
 813  
 814  
 815  
 816  
 817  
 818  
 819  
 820  
 821  
 822  
 823  
 824  
 825  
 826  
 827  
 828  
 829  
 830  
 831  
 832  
 833  
 834  
 835  
 836  
 837  
 838  
 839  
 840  
 841  
 842  
 843  
 844  
 845  
 846  
 847  
 848  
 849  
 850  
 851  
 852  
 853  
 854  
 855  
 856  
 857  
 858  
 859  
 860  
 861  
 862  
 863  
 864  
 865  
 866  
 867  
 868  
 869  
 870  
 871  
 872  
 873  
 874  
 875  
 876  
 877  
 878  
 879  
 880  
 881  
 882  
 883  
 884  
 885  
 886  
 887  
 888  
 889  
 890  
 891  
 892  
 893  
 894  
 895  
 896  
 897  
 898  
 899  
 900  
 901  
 902  
 903  
 904  
 905  
 906  
 907  
 908  
 909  
 910  
 911  
 912  
 913  
 914  
 915  
 916  
 917  
 918  
 919  
 920  
 921  
 922  
 923  
 924  
 925  
 926  
 927  
 928  
 929  
 930  
 931  
 932  
 933  
 934  
 935  
 936  
 937  
 938  
 939  
 940  
 941  
 942  
 943  
 944  
 945  
 946  
 947  
 948  
 949  
 950  
 951  
 952  
 953  
 954  
 955  
 956  
 957  
 958  
 959  
 960  
 961  
 962  
 963  
 964  
 965  
 966  
 967  
 968  
 969  
 970  
 971  
 972  
 973  
 974  
 975  
 976  
 977  
 978  
 979  
 980  
 981  
 982  
 983  
 984  
 985  
 986  
 987  
 988  
 989  
 990  
 991  
 992  
 993  
 994  
 995  
 996  
 997  
 998  
 999  
 1000  
 1001  
 1002  
 1003  
 1004  
 1005  
 1006  
 1007  
 1008  
 1009  
 1010  
 1011  
 1012  
 1013  
 1014  
 1015  
 1016  
 1017  
 1018  
 1019  
 1020  
 1021  
 1022  
 1023  
 1024  
 1025  
 1026  
 1027  
 1028  
 1029  
 1030  
 1031  
 1032  
 1033  
 1034  
 1035  
 1036  
 1037  
 1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079  
 1080  
 1081  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092  
 1093  
 1094  
 1095  
 1096  
 1097  
 1098  
 1099  
 1100  
 1101  
 1102  
 1103  
 1104  
 1105  
 1106  
 1107  
 1108  
 1109  
 1110  
 1111  
 1112  
 1113  
 1114  
 1115  
 1116  
 1117  
 1118  
 1119  
 1120  
 1121  
 1122  
 1123  
 1124  
 1125  
 1126  
 1127  
 1128  
 1129  
 1130  
 1131  
 1132  
 1133  
 1134  
 1135  
 1136  
 1137  
 1138  
 1139  
 1140  
 1141  
 1142  
 1143  
 1144  
 1145  
 1146  
 1147  
 1148  
 1149  
 1150  
 1151  
 1152  
 1153  
 1154  
 1155  
 1156  
 1157  
 1158  
 1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166  
 1167  
 1168  
 1169  
 1170  
 1171  
 1172  
 1173  
 1174  
 1175  
 1176  
 1177  
 1178  
 1179  
 1180  
 1181  
 1182  
 1183  
 1184  
 1185  
 1186  
 1187  
 1188  
 1189  
 1190  
 1191  
 1192  
 1193  
 1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238  
 1239  
 1240  
 1241  
 1242  
 1243  
 1244  
 1245  
 1246  
 1247  
 1248  
 1249  
 1250  
 1251  
 1252  
 1253  
 1254  
 1255  
 1256  
 1257  
 1258  
 1259  
 1260  
 1261  
 1262  
 1263  
 1264  
 1265  
 1266  
 1267  
 1268  
 1269  
 1270  
 1271  
 1272  
 1273  
 1274  
 1275  
 1276  
 1277  
 1278  
 1279  
 1280  
 1281  
 1282  
 1283  
 1284  
 1285  
 1286  
 1287  
 1288  
 1289  
 1290  
 1291  
 1292  
 1293  
 1294  
 1295  
 1296  
 1297  
 1298  
 1299  
 1300  
 1301  
 1302  
 1303  
 1304  
 1305  
 1306  
 1307  
 1308  
 1309  
 1310  
 1311  
 1312  
 1313  
 1314  
 1315  
 1316  
 1317  
 1318  
 1319  
 1320  
 1321  
 1322  
 1323  
 1324  
 1325  
 1326  
 1327  
 1328  
 1329  
 1330  
 1331  
 1332  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362  
 1363  
 1364  
 1365  
 1366  
 1367  
 1368  
 1369  
 1370  
 1371  
 1372  
 1373  
 1374  
 1375  
 1376  
 1377  
 1378  
 1379  
 1380  
 1381  
 1382  
 1383  
 1384  
 1385  
 1386  
 1387  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396  
 1397  
 1398  
 1399  
 1400  
 1401  
 1402  
 1403  
 1404  
 1405  
 1406  
 1407  
 1408  
 1409  
 1410  
 1411  
 1412  
 1413  
 1414  
 1415  
 1416  
 1417  
 1418  
 1419  
 1420  
 1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428  
 1429  
 1430  
 1431  
 1432  
 1433  
 1434  
 1435  
 1436  
 1437  
 1438  
 1439  
 1440  
 1441  
 1442  
 1443  
 1444  
 1445  
 1446  
 1447  
 1448  
 1449  
 1450  
 1451  
 1452  
 1453  
 1454  
 1455  
 1456  
 1457  
 1458  
 1459  
 1460  
 1461  
 1462  
 1463  
 1464  
 1465  
 1466  
 1467  
 1468  
 1469  
 1470  
 1471  
 1472  
 1473  
 1474  
 1475  
 1476  
 1477  
 1478  
 1479  
 1480  
 1481  
 1482  
 1483  
 1484  
 1485  
 1486

## 6. Overfitting on CIFAR-10

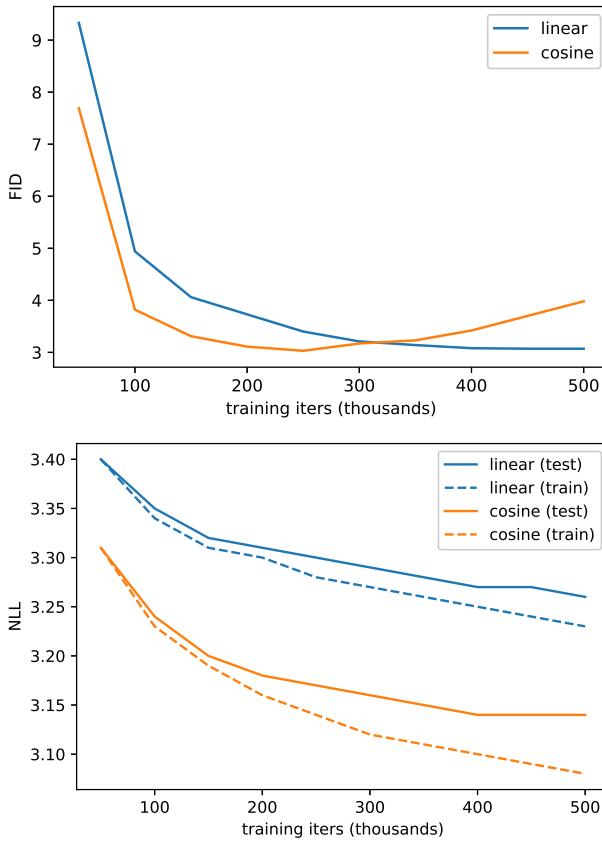


Figure 6. FID (top) and NLL (bottom) over the course of training for two CIFAR-10 models, both with dropout 0.1. The model trained with the linear schedule learns more slowly, but does not overfit as quickly. When too much overfitting occurs, we observed overfitting artifacts similar to those from Salimans et al. (2017), which is reflected by increasing FID.

All of our CIFAR-10 models experienced overfitting. This tended to hurt FID rather than producing exact training images, similar to observations from Salimans et al. (2017) and Ho et al. (2020). Before overfitting, all of our models tended to reach similar optimal FID at some point during training. Holding dropout constant, we found that models trained with our cosine schedule tended to reach optimal performance (and then overfit) more quickly than those trained with the linear schedule (Figure 6). In our experiments, we corrected for this difference by using more dropout for our cosine models than the linear models. We suspect that the overfitting from the cosine schedule is either due to 1) less noise in the cosine schedule providing less regularization, or 2) the cosine schedule making optimization, and thus overfitting, easier.

## 7. Early stopping for FID

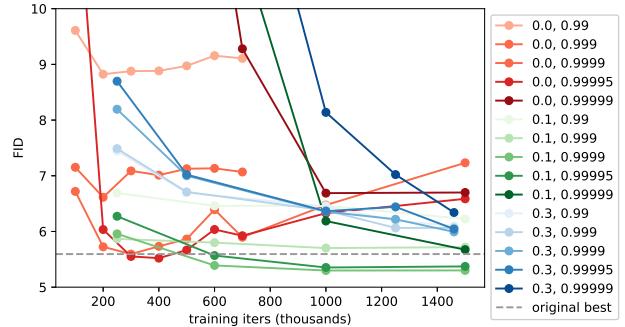


Figure 7. A sweep of dropout and EMA hyperparameters on class conditional ImageNet-64.

Like on CIFAR-10, we surprisingly observed overfitting on class-conditional ImageNet  $64 \times 64$ , despite it being a much larger and more diverse dataset. The main observable result of this overfitting was that FID started becoming worse over the course of training. We initially tried a sweep (Figure 7) over the EMA hyperparameter to make sure it was well tuned, and found that 0.9999 and 0.99995 worked best. We then tried runs with dropout 0.1 and 0.3, and found that models with a small amount of dropout improved the best attainable FID but took longer to get to the same performance and still eventually overfit. We concluded that the best way to train, given what we know, is to early stop and instead increase model size if we want to use additional training compute.

## 8. Samples with Varying Steps and Objectives

Figures 8 through 13 show unconditional ImageNet  $64 \times 64$  samples as we reduce number of sampling steps for an  $L_{\text{hybrid}}$  model with 4K diffusion steps trained for 1.5M training iterations.

Figures 14 through 19 show unconditional CIFAR-10 samples as we reduce number of sampling steps for an  $L_{\text{hybrid}}$  model with 4K diffusion steps trained for 500K training iterations.

Figures 20 and 21 highlight the difference in sample quality between models trained with  $L_{\text{hybrid}}$  and  $L_{\text{vib}}$ .



Figure 8. 50 sampling steps on unconditional ImageNet  $64 \times 64$



Figure 9. 100 sampling steps on unconditional ImageNet  $64 \times 64$

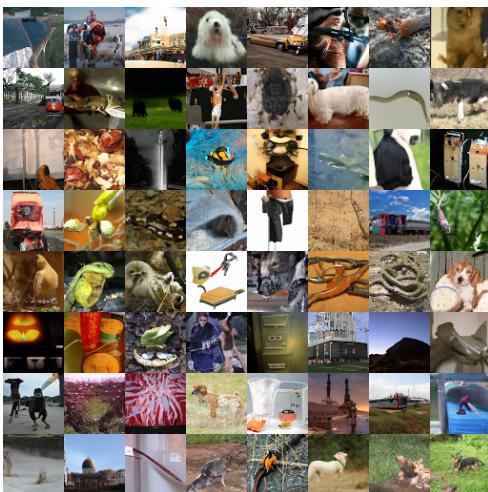


Figure 10. 200 sampling steps on unconditional ImageNet  $64 \times 64$

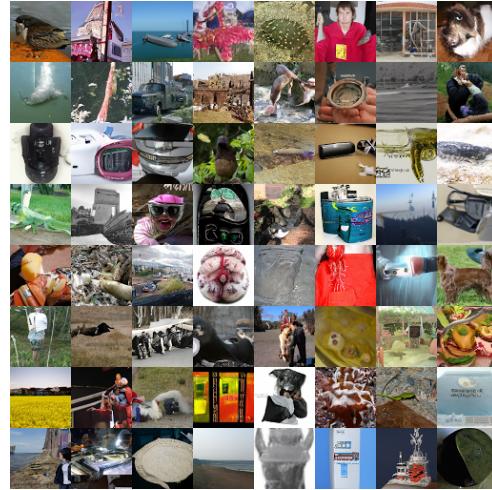


Figure 11. 400 sampling steps on unconditional ImageNet  $64 \times 64$

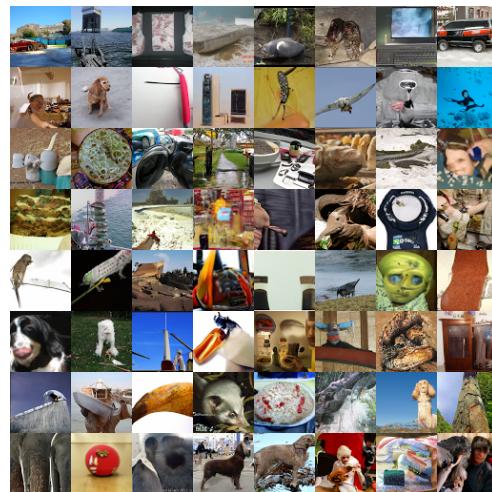


Figure 12. 1000 sampling steps on unconditional ImageNet  $64 \times 64$

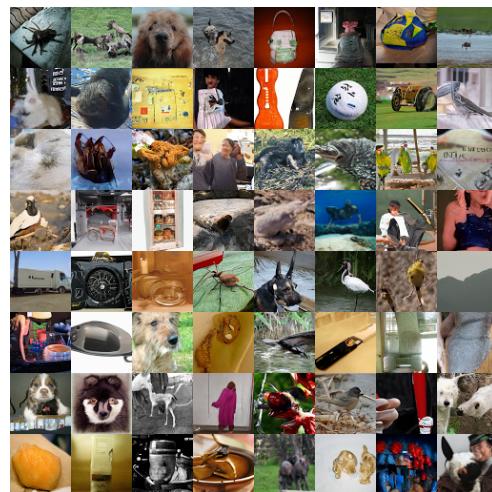


Figure 13. 4K sampling steps on unconditional ImageNet  $64 \times 64$ .

220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235

236 Figure 8. 50 sampling steps on unconditional ImageNet  $64 \times 64$   
237

238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254

255 Figure 9. 100 sampling steps on unconditional ImageNet  $64 \times 64$   
256

257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272

273 Figure 10. 200 sampling steps on unconditional ImageNet  $64 \times 64$   
274

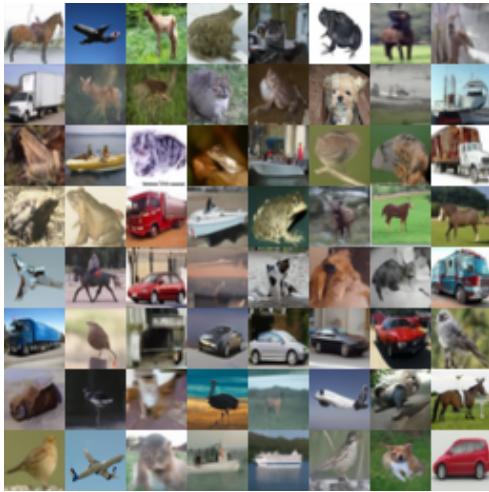


Figure 14. 50 sampling steps on unconditional CIFAR-10

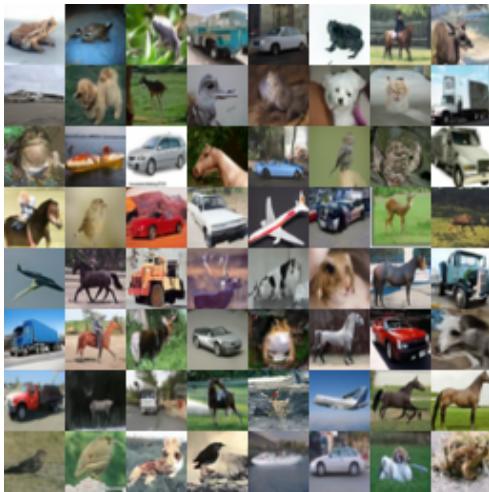


Figure 15. 100 sampling steps on unconditional CIFAR-10

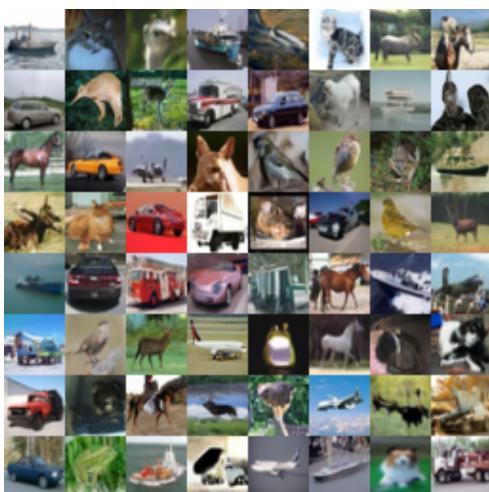


Figure 16. 200 sampling steps on unconditional CIFAR-10



Figure 17. 400 sampling steps on unconditional CIFAR-10

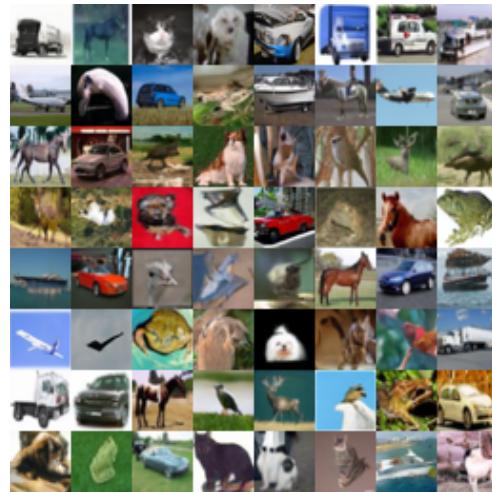


Figure 18. 1000 sampling steps on unconditional CIFAR-10



Figure 19. 4000 sampling steps on unconditional CIFAR-10

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

Figure 20. Unconditional ImageNet  $64 \times 64$  samples generated from  $L_{\text{hybrid}}$  (top) and  $L_{\text{vib}}$  (bottom) models using the exact same random noise. Both models were trained for 1.5M iterations.

372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384

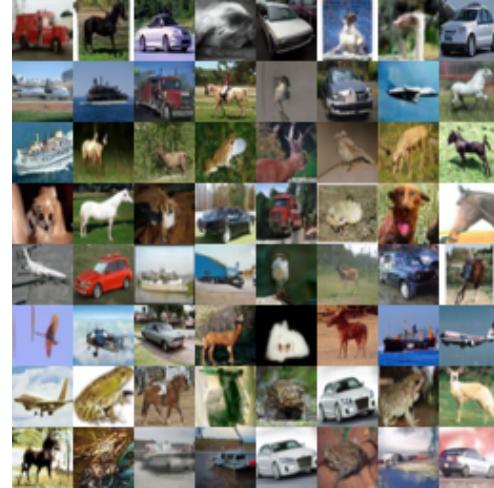


Figure 21. Unconditional CIFAR-10 samples generated from  $L_{\text{hybrid}}$  (top) and  $L_{\text{vib}}$  (bottom) models using the exact same random noise. Both models were trained for 500K iterations.

**References**

- 385  
386 Brock, A., Donahue, J., and Simonyan, K. Large scale gan  
387 training for high fidelity natural image synthesis. *arXiv*  
388 preprint *arXiv:1809.11096*, 2018.  
389  
390 He, K., Zhang, X., Ren, S., and Sun, J. Deep residual  
391 learning for image recognition, 2015.  
392  
393 Ho, J., Jain, A., and Abbeel, P. Denoising diffusion proba-  
394 bilistic models, 2020.  
395  
396 Kingma, D. P. and Ba, J. Adam: A method for stochastic  
397 optimization, 2014.  
398  
399 Ravuri, S. and Vinyals, O. Classification accuracy score  
400 for conditional generative models. *arXiv preprint*  
401 *arXiv:1905.10887*, 2019.  
402  
403 Razavi, A., van den Oord, A., and Vinyals, O. Generating  
404 diverse high-fidelity images with vq-vae-2, 2019.  
405  
406 Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pix-  
407 elcnn++: Improving the pixelcnn with discretized logistic  
408 mixture likelihood and other modifications, 2017.  
409  
410 Song, J., Meng, C., and Ermon, S. Denoising diffusion  
411 implicit models, 2020.  
412  
413 van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K.  
414 Pixel recurrent neural networks, 2016.  
415  
416 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones,  
417 L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention  
418 is all you need, 2017.  
419  
420 Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and  
421 Xiao, J. Lsun: Construction of a large-scale image dataset  
422 using deep learning with humans in the loop, 2015.  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439