```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/
performance_get_output_generator.py
[0.2734375  0.19775391 0.45996094 0.06884766]
[0.28125    0.22070312 0.43798828 0.06005859]
[0.28759766 0.18554688 0.4453125  0.08154297]
[0.26464844 0.20068359 0.47998047 0.0546875 ]
[0.27636719 0.20410156 0.45800781 0.06152344]
[0.26025391 0.23193359 0.45263672 0.05517578]
[0.26708984 0.21386719 0.45898438 0.06005859]
[0.27099609 0.22363281 0.44384766 0.06152344]
[0.27197266 0.20849609 0.45654297 0.06298828]
[0.27246094 0.22460938 0.44433594 0.05859375]
[0.28027344 0.21044922 0.44580078 0.06347656]
[0.26220703 0.21777344 0.45947266 0.06054688]
[0.26611328 0.21679688 0.45068359 0.06640625]
[0.26171875 0.21191406 0.46191406 0.06445312]
[0.27685547 0.19873047 0.45996094 0.06445312]
[0.26611328 0.20898438 0.45214844 0.07275391]
[0.27099609 0.203125   0.46044922 0.06542969]
[0.2421875  0.21191406 0.47265625 0.07324219]
[0.2734375  0.21142578 0.45703125 0.05810547]
[0.26953125 0.20996094 0.44580078 0.07470703]
[0.28320312 0.20751953 0.45117188 0.05810547]
[0.26953125 0.2109375  0.45117188 0.06835938]
[0.28417969 0.20849609 0.43945312 0.06787109]
[0.27734375 0.21142578 0.44189453 0.06933594]
[0.28369141 0.20996094 0.44335938 0.06298828]
[0.28076172 0.20458984 0.44824219 0.06640625]
[0.28417969 0.20898438 0.43896484 0.06787109]
[0.26367188 0.20947266 0.46142578 0.06542969]
[0.27148438 0.20019531 0.46777344 0.06054688]
[0.27880859 0.21582031 0.43701172 0.06835938]
[0.27441406 0.21191406 0.45410156 0.05957031]
[0.26171875 0.22167969 0.44677734 0.06982422]
[0.28515625 0.20947266 0.44287109 0.0625    ]
[0.27197266 0.20556641 0.45898438 0.06347656]
[0.26708984 0.22509766 0.44042969 0.06738281]
[0.27636719 0.21142578 0.44140625 0.07080078]
[0.27148438 0.21044922 0.43945312 0.07861328]
[0.2890625  0.20263672 0.44287109 0.06542969]
[0.30126953 0.19384766 0.43945312 0.06542969]
[0.26220703 0.22021484 0.44628906 0.07128906]
[0.29150391 0.21191406 0.43847656 0.05810547]
[0.25927734 0.21044922 0.45996094 0.0703125 ]
[0.27294922 0.20703125 0.45556641 0.06445312]
[0.28613281 0.20410156 0.45166016 0.05810547]
[0.27490234 0.21972656 0.44482422 0.06054688]
[0.28271484 0.21533203 0.43554688 0.06640625]
[0.27001953 0.22021484 0.44091797 0.06884766]
[0.26318359 0.20947266 0.45751953 0.06982422]
[0.28369141 0.20898438 0.43994141 0.06738281]
[0.27050781 0.19628906 0.46142578 0.07177734]
[0.27783203 0.21435547 0.43847656 0.06933594]
[0.25634766 0.21582031 0.4609375  0.06689453]
[0.27734375 0.20410156 0.45654297 0.06201172]
[0.27197266 0.20166016 0.4609375  0.06542969]
[0.27001953 0.203125   0.46630859 0.06054688]
[0.27832031 0.19580078 0.45947266 0.06640625]
[0.28027344 0.21386719 0.43359375 0.07226562]
[0.27050781 0.21630859 0.44921875 0.06396484]
[0.28027344 0.20361328 0.44970703 0.06640625]
[0.27197266 0.20166016 0.45117188 0.07519531]
[0.265625   0.19921875 0.45947266 0.07568359]
[0.28173828 0.19482422 0.46289062 0.06054688]
[0.27539062 0.22119141 0.44238281 0.06103516]
[0.28027344 0.19238281 0.46533203 0.06201172]
[0.27929688 0.21289062 0.43457031 0.07324219]
[0.27587891 0.20751953 0.45410156 0.0625    ]
[0.26416016 0.20800781 0.45898438 0.06884766]
[0.28271484 0.21386719 0.43310547 0.0703125 ]
[0.26757812 0.20898438 0.45410156 0.06933594]
[0.28125    0.20751953 0.45751953 0.05371094]
[0.28662109 0.19287109 0.45898438 0.06152344]
[0.28613281 0.19970703 0.45068359 0.06347656]
[0.28515625 0.20361328 0.44433594 0.06689453]
[0.27734375 0.21484375 0.44140625 0.06640625]
[0.26513672 0.21630859 0.45410156 0.06445312]
[0.30273438 0.21484375 0.42480469 0.05761719]
[0.27001953 0.20605469 0.45849609 0.06542969]
[0.26660156 0.21191406 0.46191406 0.05957031]
[0.26904297 0.20361328 0.44921875 0.078125  ]
[0.26025391 0.21044922 0.45898438 0.0703125 ]
[0.25390625 0.20263672 0.47509766 0.06835938]
[0.27880859 0.19970703 0.45410156 0.06738281]
[0.27636719 0.20751953 0.45703125 0.05908203]
[0.28564453 0.21679688 0.43115234 0.06640625]
[0.27929688 0.20166016 0.46191406 0.05712891]
[0.26904297 0.22070312 0.45019531 0.06005859]
[0.27001953 0.20654297 0.45605469 0.06738281]
```

```
[0.28759766 0.21923828 0.42626953 0.06689453]
[0.27685547 0.19091797 0.46240234 0.06982422]
[0.26513672 0.20751953 0.46240234 0.06494141]
[0.28808594 0.19921875 0.44482422 0.06787109]
[0.24853516 0.23095703 0.44921875 0.07128906]
[0.28320312 0.20898438 0.44873047 0.05908203]
[0.25634766 0.21777344 0.46240234 0.06347656]
[0.28076172 0.1875     0.47363281 0.05810547]
[0.25927734 0.21337891 0.46923828 0.05810547]
[0.27832031 0.19189453 0.45605469 0.07373047]
[0.27929688 0.21826172 0.44482422 0.05761719]
[0.2734375  0.21337891 0.44091797 0.07226562]
[0.28466797 0.22167969 0.42773438 0.06591797]
[0.27636719 0.20214844 0.45800781 0.06347656]
[0.25634766 0.20605469 0.47119141 0.06640625]
[0.27441406 0.20556641 0.44970703 0.0703125 ]
[0.28173828 0.20361328 0.44726562 0.06738281]
[0.27050781 0.21533203 0.44189453 0.07226562]
[0.28320312 0.19873047 0.44628906 0.07177734]
[0.28369141 0.19873047 0.45458984 0.06298828]
[0.28466797 0.19433594 0.45751953 0.06347656]
[0.26171875 0.20019531 0.47119141 0.06689453]
[0.29150391 0.20849609 0.43310547 0.06689453]
[0.27636719 0.20996094 0.45019531 0.06347656]
[0.28027344 0.19775391 0.44140625 0.08056641]
[0.28515625 0.21777344 0.43554688 0.06152344]
[0.28955078 0.19335938 0.45507812 0.06201172]
[0.26220703 0.20263672 0.46337891 0.07177734]
[0.24951172 0.22753906 0.45556641 0.06738281]
[0.26757812 0.20703125 0.46337891 0.06201172]
[0.25537109 0.22607422 0.45117188 0.06738281]
[0.26953125 0.19335938 0.47265625 0.06445312]
[0.26953125 0.19873047 0.46777344 0.06396484]
[0.27441406 0.21191406 0.44384766 0.06982422]
[0.28515625 0.21484375 0.43701172 0.06298828]
[0.26269531 0.20996094 0.44726562 0.08007812]
[0.25439453 0.22460938 0.45605469 0.06494141]
[0.27636719 0.22412109 0.44091797 0.05859375]
[0.25927734 0.20166016 0.46972656 0.06933594]
[0.26708984 0.21289062 0.45214844 0.06787109]
[0.28710938 0.20605469 0.43164062 0.07519531]
[0.27099609 0.20849609 0.45068359 0.06982422]
[0.27148438 0.21142578 0.45019531 0.06689453]
[0.27050781 0.20410156 0.46533203 0.06005859]
[0.27148438 0.21679688 0.45263672 0.05908203]
[0.27001953 0.20703125 0.46191406 0.06103516]
[0.26513672 0.19042969 0.47460938 0.06982422]
[0.26904297 0.22753906 0.44042969 0.06298828]
[0.265625   0.18603516 0.47705078 0.07128906]
[0.26855469 0.21972656 0.44482422 0.06689453]
[0.26416016 0.20117188 0.45996094 0.07470703]
[0.28466797 0.20458984 0.45166016 0.05908203]
[0.27148438 0.21289062 0.44921875 0.06640625]
[0.29101562 0.20947266 0.44091797 0.05859375]
[0.27587891 0.1875     0.46435547 0.07226562]
[0.26757812 0.20166016 0.46044922 0.0703125 ]
[0.26464844 0.21923828 0.45410156 0.06201172]
[0.27587891 0.21875    0.43945312 0.06591797]
[0.29052734 0.19677734 0.44775391 0.06494141]
[0.27539062 0.19091797 0.47070312 0.06298828]
[0.27685547 0.19726562 0.45996094 0.06591797]
[0.26855469 0.20800781 0.44726562 0.07617188]
[0.26660156 0.22216797 0.45068359 0.06054688]
[0.28027344 0.19091797 0.45214844 0.07666016]
[0.26464844 0.20800781 0.46582031 0.06152344]
[0.26318359 0.19677734 0.48291016 0.05712891]
[0.27880859 0.19873047 0.45947266 0.06298828]
[0.26171875 0.20654297 0.45800781 0.07373047]
[0.26464844 0.20166016 0.4765625  0.05712891]
[0.27392578 0.20117188 0.46777344 0.05712891]
[0.26464844 0.19873047 0.46777344 0.06884766]
[0.28222656 0.21777344 0.43066406 0.06933594]
[0.27392578 0.20410156 0.45458984 0.06738281]
[0.26904297 0.20507812 0.46679688 0.05908203]
[0.28125    0.21044922 0.44677734 0.06152344]
[0.28613281 0.20263672 0.44433594 0.06689453]
[0.27050781 0.2109375  0.45117188 0.06738281]
[0.25732422 0.22851562 0.44335938 0.07080078]
[0.28515625 0.20556641 0.44628906 0.06298828]
[0.27685547 0.21728516 0.43798828 0.06787109]
[0.25048828 0.23339844 0.44677734 0.06933594]
[0.2734375  0.20019531 0.45703125 0.06933594]
[0.24755859 0.20898438 0.47851562 0.06494141]
[0.28125    0.18896484 0.46435547 0.06542969]
[0.27783203 0.20996094 0.44287109 0.06933594]
[0.27099609 0.21679688 0.44677734 0.06542969]
[0.27880859 0.21826172 0.44384766 0.05908203]
[0.29589844 0.203125   0.43115234 0.06982422]
[0.27197266 0.20849609 0.45703125 0.0625    ]
```

```
[0.26904297 0.21191406 0.45751953 0.06152344]
[0.28320312 0.20898438 0.44775391 0.06005859]
[0.27539062 0.20849609 0.44335938 0.07275391]
[0.26806641 0.21337891 0.45703125 0.06152344]
[0.2734375  0.203125   0.45898438 0.06445312]
[0.28271484 0.20751953 0.43945312 0.0703125 ]
[0.27783203 0.21777344 0.43164062 0.07275391]
[0.29199219 0.20947266 0.43261719 0.06591797]
[0.26660156 0.21240234 0.45800781 0.06298828]
[0.27001953 0.20117188 0.46142578 0.06738281]
[0.28076172 0.20751953 0.44824219 0.06347656]
[0.26611328 0.21435547 0.45703125 0.0625    ]
[0.27294922 0.21679688 0.44873047 0.06152344]
[0.28955078 0.20849609 0.43457031 0.06738281]
[0.27929688 0.19873047 0.45263672 0.06933594]
[0.28613281 0.20507812 0.44384766 0.06494141]
[0.26953125 0.22314453 0.44970703 0.05761719]
[0.27392578 0.21533203 0.44140625 0.06933594]
[0.27929688 0.18994141 0.46875    0.06201172]
[0.26416016 0.22119141 0.45263672 0.06201172]
[0.26464844 0.22900391 0.44433594 0.06201172]
[0.29150391 0.20947266 0.43164062 0.06738281]
[0.26660156 0.22607422 0.4375     0.06982422]
[0.26220703 0.22314453 0.44189453 0.07275391]
Timer unit: 1e-07 s

Total time: 14.9875 s
File: C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_get_output_generator.py
Function: get_output_V1 at line 34

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    34                                           def get_output_V1():
    35       101       2767.0     27.4      0.0      for noise in batch_noise:
    36       100       1833.0     18.3      0.0          real_keys = {"00", "10", "01", "11"}
    37
    38       100      77501.0    775.0      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    39       100     210480.0   2104.8      0.1          qc = QuantumCircuit(sum(num_qubits))
    40
    41       100     140614.0   1406.1      0.1          init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    42       100       2875.0     28.8      0.0          assert noise.shape[0] == sum(num_qubits)
    43
    44       300       4800.0     16.0      0.0          for num_qubit in range(sum(num_qubits)):
    45       200     218054.0   1090.3      0.1              init_dist.ry(noise[num_qubit], num_qubit)
    46
    47       100       4036.0     40.4      0.0          params = cast(np.ndarray, parameter_values)
    48
    49       100   11977211.0 119772.1      8.0          qc.append(construct_circuit(params), quantum)
    50       100    3650587.0  36505.9      2.4          final_circuit = qc.compose(init_dist, front=True)
    51       100     332303.0   3323.0      0.2          final_circuit.measure_all()
    52
    53       100    4408498.0  44085.0      2.9          simulator_1 = qiskit.Aer.get_backend("aer_simulator")
    54       100  119741433.0 1197414.3     79.9          final_circuit = qiskit.transpile(final_circuit, simulator_1)
    55       100    8256139.0  82561.4      5.5          result = simulator_1.run(final_circuit, shots=shots).result()
    56       100     194864.0   1948.6      0.1          counts = result.get_counts(final_circuit)
    57
    58       100       1397.0     14.0      0.0          try:
    59       100      23936.0    239.4      0.0              pixels = np.array([counts["00"], counts["10"], counts["01"], counts["11"]])
    60
    61                                                   except KeyError:
    62                                                       # dealing with the keys that qiskit doesn't include in the
    63                                                       # dictionary because they don't get any measurements
    64
    65                                                       keys = counts.keys()
    66                                                       missing_keys = real_keys.difference(keys)
    67                                                       # we use sets to get the missing keys
    68                                                       for key_missing in missing_keys:
    69                                                           counts[key_missing] = 0
    70
    71                                                       pixels = np.array([counts["00"], counts["10"], counts["01"], counts["11"]])
    72
    73       100      31247.0    312.5      0.0          pixels = pixels / shots
    74       100     594056.0   5940.6      0.4          print(pixels)

Total time: 12.6999 s
File: C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_get_output_generator.py
Function: get_output_V2 at line 77

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    77                                           def get_output_V2():
    78         1       5399.0   5399.0      0.0      simulator = qiskit.Aer.get_backend("aer_simulator")
    79       101       3298.0     32.7      0.0      for noise in batch_noise:
    80       100       2276.0     22.8      0.0          real_keys = {"00", "10", "01", "11"}
    81
    82       100      90775.0    907.8      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    83       100     238127.0   2381.3      0.2          qc = QuantumCircuit(sum(num_qubits))
```

```
    84
    85       100     154557.0   1545.6      0.1         init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    86       100       3432.0     34.3      0.0         assert noise.shape[0] == sum(num_qubits)
    87
    88       300       4839.0     16.1      0.0         for num_qubit in range(sum(num_qubits)):
    89       200     221874.0   1109.4      0.2             init_dist.ry(noise[num_qubit], num_qubit)
    90
    91       100       4874.0     48.7      0.0         params = cast(np.ndarray, parameter_values)
    92
    93       100   11485442.0 114854.4      9.0         qc.append(construct_circuit(params), quantum)
    94       100    3475603.0  34756.0      2.7         final_circuit = qc.compose(init_dist, front=True)
    95       100     322252.0   3222.5      0.3         final_circuit.measure_all()
    96
    97       100  101968788.0 1019687.9     80.3         final_circuit = qiskit.transpile(final_circuit, simulator)
    98       100    8110570.0  81105.7      6.4         result = simulator.run(final_circuit, shots=shots).result()
    99       100     206713.0   2067.1      0.2         counts = result.get_counts(final_circuit)
   100
   101       100       1452.0     14.5      0.0         try:
   102       100      26541.0    265.4      0.0             pixels = np.array([counts["00"], counts["10"], counts["01"], counts["11"]])
   103
   104                                                 except KeyError:
   105                                                     # dealing with the keys that qiskit doesn't include in the
   106                                                     # dictionary because they don't get any measurements
   107
   108                                                     keys = counts.keys()
   109                                                     missing_keys = real_keys.difference(keys)
   110                                                     # we use sets to get the missing keys
   111                                                     for key_missing in missing_keys:
   112                                                         counts[key_missing] = 0
   113
   114                                                     pixels = np.array([counts["00"], counts["10"], counts["01"], counts["11"]])
   115
   116       100      35404.0    354.0      0.0         pixels = pixels / shots
   117       100     637005.0   6370.1      0.5         print(pixels)


Process finished with exit code 0
```