```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/
performance_get_output_generator.py
[0.20996094 0.34033203 0.42773438 0.02197266]
[0.22460938 0.35742188 0.39990234 0.01806641]
[0.20556641 0.35351562 0.42333984 0.01757812]
[0.21923828 0.34277344 0.42041016 0.01757812]
[0.20703125 0.35791016 0.41601562 0.01904297]
[0.21044922 0.35498047 0.41162109 0.02294922]
[0.20800781 0.36279297 0.40722656 0.02197266]
[0.21679688 0.36425781 0.40136719 0.01757812]
[0.203125   0.35009766 0.42626953 0.02050781]
[0.21142578 0.36132812 0.41210938 0.01513672]
[0.21777344 0.35644531 0.41064453 0.01513672]
[0.20800781 0.35205078 0.42285156 0.01708984]
[0.20849609 0.35009766 0.41796875 0.0234375 ]
[0.20556641 0.33300781 0.43896484 0.02246094]
[0.21728516 0.34130859 0.42285156 0.01855469]
[0.20800781 0.34667969 0.42480469 0.02050781]
[0.22070312 0.34765625 0.41601562 0.015625  ]
[0.18359375 0.38330078 0.41748047 0.015625  ]
[0.21289062 0.33007812 0.43066406 0.02636719]
[0.21337891 0.35498047 0.41455078 0.01708984]
[0.19726562 0.36132812 0.42529297 0.01611328]
[0.20751953 0.35693359 0.41552734 0.02001953]
[0.22119141 0.34667969 0.41162109 0.02050781]
[0.20703125 0.34179688 0.43701172 0.01416016]
[0.22412109 0.33642578 0.421875   0.01757812]
[0.20263672 0.35400391 0.42333984 0.02001953]
[0.22265625 0.32324219 0.43310547 0.02099609]
[0.21777344 0.35498047 0.40820312 0.01904297]
[0.20605469 0.35009766 0.42822266 0.015625  ]
[0.20556641 0.36132812 0.41748047 0.015625  ]
[0.21289062 0.35644531 0.41162109 0.01904297]
[0.20849609 0.35351562 0.42138672 0.01660156]
[0.21679688 0.34228516 0.42578125 0.01513672]
[0.22070312 0.35205078 0.40820312 0.01904297]
[0.19775391 0.3671875  0.41601562 0.01904297]
[0.22509766 0.328125   0.42871094 0.01806641]
[0.2109375  0.35107422 0.41894531 0.01904297]
[0.21533203 0.34619141 0.42041016 0.01806641]
[0.22216797 0.34765625 0.41015625 0.02001953]
[0.21972656 0.34960938 0.41455078 0.01611328]
[0.20458984 0.35986328 0.4140625  0.02148438]
[0.22021484 0.35791016 0.40576172 0.01611328]
[0.21044922 0.35546875 0.41699219 0.01708984]
[0.19873047 0.36230469 0.41992188 0.01904297]
[0.23925781 0.35400391 0.38476562 0.02197266]
[0.20117188 0.359375   0.42285156 0.01660156]
[0.20751953 0.36328125 0.4140625  0.01513672]
[0.20605469 0.35107422 0.42236328 0.02050781]
[0.203125   0.37207031 0.40625    0.01855469]
[0.20654297 0.34472656 0.42822266 0.02050781]
[0.21337891 0.35546875 0.41259766 0.01855469]
[0.21679688 0.35107422 0.41210938 0.02001953]
[0.23388672 0.33398438 0.41162109 0.02050781]
[0.21728516 0.33837891 0.42675781 0.01757812]
[0.19726562 0.38525391 0.39794922 0.01953125]
[0.22021484 0.34912109 0.40820312 0.02246094]
[0.22460938 0.32421875 0.43261719 0.01855469]
[0.22167969 0.34912109 0.40966797 0.01953125]
[0.2265625  0.34130859 0.41796875 0.01416016]
[0.20507812 0.34472656 0.43310547 0.01708984]
[0.21337891 0.35351562 0.4140625  0.01904297]
[0.20751953 0.35058594 0.42333984 0.01855469]
[0.22753906 0.32666016 0.42480469 0.02099609]
[0.20166016 0.34082031 0.4375     0.02001953]
[0.20654297 0.3515625  0.42480469 0.01708984]
[0.21679688 0.35644531 0.40869141 0.01806641]
[0.20605469 0.34960938 0.42431641 0.02001953]
[0.20361328 0.36669922 0.41601562 0.01367188]
[0.22119141 0.35302734 0.40820312 0.01757812]
[0.20947266 0.34667969 0.42724609 0.01660156]
[0.21044922 0.34912109 0.421875   0.01855469]
[0.21972656 0.33886719 0.42285156 0.01855469]
[0.21386719 0.35009766 0.42236328 0.01367188]
[0.19482422 0.33691406 0.44824219 0.02001953]
[0.20751953 0.35009766 0.41894531 0.0234375 ]
[0.20166016 0.36328125 0.41650391 0.01855469]
[0.21044922 0.35693359 0.40966797 0.02294922]
[0.19970703 0.37207031 0.41210938 0.01611328]
[0.21972656 0.34814453 0.41357422 0.01855469]
[0.22021484 0.3515625  0.41259766 0.015625  ]
[0.21240234 0.34423828 0.42675781 0.01660156]
[0.21533203 0.35986328 0.40722656 0.01757812]
[0.19140625 0.34667969 0.43847656 0.0234375 ]
[0.20068359 0.36328125 0.41796875 0.01806641]
[0.22021484 0.34912109 0.40869141 0.02197266]
[0.22314453 0.33544922 0.42480469 0.01660156]
[0.21777344 0.34228516 0.41796875 0.02197266]
```

```
[0.21728516 0.36279297 0.40039062 0.01953125]
[0.2109375  0.35253906 0.41552734 0.02099609]
[0.19677734 0.36767578 0.41796875 0.01757812]
[0.22607422 0.36376953 0.39501953 0.01513672]
[0.20019531 0.36425781 0.42285156 0.01269531]
[0.21923828 0.33984375 0.41943359 0.02148438]
[0.21582031 0.33349609 0.43017578 0.02050781]
[0.22705078 0.36425781 0.39111328 0.01757812]
[0.19384766 0.34667969 0.43554688 0.02392578]
[0.20458984 0.34570312 0.43261719 0.01708984]
[0.20507812 0.34619141 0.43212891 0.01660156]
[0.22216797 0.3359375  0.42382812 0.01806641]
[0.21679688 0.33398438 0.42724609 0.02197266]
[0.21533203 0.3515625  0.41894531 0.01416016]
[0.21337891 0.35644531 0.41308594 0.01708984]
[0.19873047 0.35791016 0.42089844 0.02246094]
[0.21337891 0.33837891 0.43359375 0.01464844]
[0.20751953 0.34765625 0.42480469 0.02001953]
[0.22753906 0.32958984 0.42041016 0.02246094]
[0.19970703 0.36279297 0.41601562 0.02148438]
[0.20849609 0.35205078 0.41601562 0.0234375 ]
[0.18847656 0.35986328 0.43359375 0.01806641]
[0.20654297 0.34765625 0.42529297 0.02050781]
[0.21972656 0.34765625 0.41455078 0.01806641]
[0.21875    0.359375   0.40332031 0.01855469]
[0.2265625  0.3515625  0.40283203 0.01904297]
[0.21630859 0.34765625 0.41015625 0.02587891]
[0.23339844 0.34814453 0.3984375  0.02001953]
[0.21142578 0.35839844 0.41210938 0.01806641]
[0.22167969 0.33300781 0.42675781 0.01855469]
[0.21142578 0.35693359 0.41455078 0.01708984]
[0.22851562 0.32275391 0.42773438 0.02099609]
[0.20361328 0.34521484 0.42919922 0.02197266]
[0.20654297 0.359375   0.41796875 0.01611328]
[0.23388672 0.34130859 0.40673828 0.01806641]
[0.21142578 0.34863281 0.42333984 0.01660156]
[0.2265625  0.34228516 0.41015625 0.02099609]
[0.21435547 0.36083984 0.40820312 0.01660156]
[0.20410156 0.36035156 0.41845703 0.01708984]
[0.21435547 0.36279297 0.39941406 0.0234375 ]
[0.22558594 0.34716797 0.40576172 0.02148438]
[0.21533203 0.359375   0.40380859 0.02148438]
[0.22070312 0.34765625 0.40869141 0.02294922]
[0.20751953 0.36621094 0.40380859 0.02246094]
[0.20605469 0.36767578 0.40380859 0.02246094]
[0.20947266 0.34619141 0.42773438 0.01660156]
[0.20996094 0.36083984 0.40869141 0.02050781]
[0.21484375 0.35693359 0.41552734 0.01269531]
[0.20947266 0.35400391 0.4140625  0.02246094]
[0.20605469 0.35449219 0.42236328 0.01708984]
[0.22167969 0.3359375  0.42529297 0.01708984]
[0.21289062 0.35351562 0.41503906 0.01855469]
[0.20849609 0.34130859 0.4296875  0.02050781]
[0.21582031 0.34033203 0.41943359 0.02441406]
[0.20166016 0.35009766 0.42626953 0.02197266]
[0.20800781 0.36181641 0.41552734 0.01464844]
[0.22119141 0.33056641 0.42919922 0.01904297]
[0.20654297 0.35595703 0.41308594 0.02441406]
[0.20898438 0.35546875 0.41455078 0.02099609]
[0.22314453 0.33789062 0.42431641 0.01464844]
[0.19384766 0.35498047 0.43261719 0.01855469]
[0.21142578 0.359375   0.40869141 0.02050781]
[0.20507812 0.36132812 0.41259766 0.02099609]
[0.22119141 0.34912109 0.41113281 0.01855469]
[0.20263672 0.35546875 0.42236328 0.01953125]
[0.21679688 0.33691406 0.43115234 0.01513672]
[0.19824219 0.34960938 0.42236328 0.02978516]
[0.23144531 0.34619141 0.40771484 0.01464844]
[0.22119141 0.35449219 0.40478516 0.01953125]
[0.21484375 0.36816406 0.39794922 0.01904297]
[0.21044922 0.35009766 0.42675781 0.01269531]
[0.19335938 0.35839844 0.42578125 0.02246094]
[0.20947266 0.35449219 0.41601562 0.02001953]
[0.21142578 0.33691406 0.43652344 0.01513672]
[0.20361328 0.35009766 0.42675781 0.01953125]
[0.21142578 0.32910156 0.44091797 0.01855469]
[0.19921875 0.35644531 0.42822266 0.01611328]
[0.23339844 0.34033203 0.41210938 0.01416016]
[0.22070312 0.32666016 0.43261719 0.02001953]
[0.20947266 0.35498047 0.421875   0.01367188]
[0.22460938 0.35058594 0.40478516 0.02001953]
[0.21875    0.35107422 0.41308594 0.01708984]
[0.21679688 0.35498047 0.40917969 0.01904297]
[0.20849609 0.35009766 0.42236328 0.01904297]
[0.20410156 0.35400391 0.41894531 0.02294922]
[0.20507812 0.36425781 0.41064453 0.02001953]
[0.22021484 0.33984375 0.41894531 0.02099609]
[0.20703125 0.34570312 0.43115234 0.01611328]
[0.21777344 0.35253906 0.4140625  0.015625  ]
```

```
[0.20117188 0.37109375 0.41210938 0.015625  ]
[0.21191406 0.35107422 0.41748047 0.01953125]
[0.22949219 0.3515625  0.40478516 0.01416016]
[0.22900391 0.34082031 0.40429688 0.02587891]
[0.20263672 0.33056641 0.44873047 0.01806641]
[0.20947266 0.33251953 0.44384766 0.01416016]
[0.22460938 0.36425781 0.39550781 0.015625  ]
[0.21533203 0.33984375 0.42480469 0.02001953]
[0.19873047 0.35595703 0.42626953 0.01904297]
[0.20507812 0.34326172 0.43652344 0.01513672]
[0.22753906 0.33154297 0.41796875 0.02294922]
[0.21972656 0.34570312 0.41894531 0.015625  ]
[0.21240234 0.34863281 0.42041016 0.01855469]
[0.21289062 0.36035156 0.40429688 0.02246094]
[0.2265625  0.33251953 0.42041016 0.02050781]
[0.20947266 0.32617188 0.44580078 0.01855469]
[0.21240234 0.35986328 0.41113281 0.01660156]
[0.2109375  0.35351562 0.41552734 0.02001953]
[0.21826172 0.33935547 0.42333984 0.01904297]
[0.1953125  0.359375   0.42236328 0.02294922]
[0.21240234 0.34033203 0.42822266 0.01904297]
[0.2109375  0.34130859 0.42285156 0.02490234]
[0.20605469 0.34716797 0.42871094 0.01806641]
[0.20214844 0.34326172 0.43505859 0.01953125]
Timer unit: 1e-07 s

Total time: 14.2574 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       2917.0     28.9      0.0      for noise in batch_noise:
    36       100       2037.0     20.4      0.0          real_keys = {"00", "10", "01", "11"}
    37
    38       100     113679.0   1136.8      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    39       100     289161.0   2891.6      0.2          qc = QuantumCircuit(sum(num_qubits))
    40
    41       100     147993.0   1479.9      0.1          init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    42       100       3007.0     30.1      0.0          assert noise.shape[0] == sum(num_qubits)
    43
    44       300       4681.0     15.6      0.0          for num_qubit in range(sum(num_qubits)):
    45       200     204298.0   1021.5      0.1              init_dist.ry(noise[num_qubit], num_qubit)
    46
    47       100       3705.0     37.0      0.0          params = cast(np.ndarray, parameter_values)
    48
    49       100   11779196.0 117792.0      8.3          qc.append(construct_circuit(params), quantum)
    50       100    3665232.0  36652.3      2.6          final_circuit = qc.compose(init_dist, front=True)
    51       100     408212.0   4082.1      0.3          final_circuit.measure_all()
    52
    53       100    4173729.0  41737.3      2.9          simulator_1 = qiskit.Aer.get_backend("aer_simulator")
    54       100  112788207.0 1127882.1     79.1          final_circuit = qiskit.transpile(final_circuit, simulator_1)
    55       100    8054560.0  80545.6      5.6          result = simulator_1.run(final_circuit, shots=shots).result()
    56       100     191907.0   1919.1      0.1          counts = result.get_counts(final_circuit)
    57
    58       100       1411.0     14.1      0.0          try:
    59       100      57286.0    572.9      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      38161.0    381.6      0.0          pixels = pixels / shots
    74       100     644146.0   6441.5      0.5          print(pixels)

Total time: 13.7492 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       2928.0   2928.0      0.0      simulator = qiskit.Aer.get_backend("aer_simulator")
    79       101       3235.0     32.0      0.0      for noise in batch_noise:
    80       100       2211.0     22.1      0.0          real_keys = {"00", "10", "01", "11"}
    81
    82       100     109241.0   1092.4      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    83       100     225232.0   2252.3      0.2          qc = QuantumCircuit(sum(num_qubits))
```

```
    84
    85       100     147240.0   1472.4      0.1         init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    86       100       3118.0     31.2      0.0         assert noise.shape[0] == sum(num_qubits)
    87
    88       300       4796.0     16.0      0.0         for num_qubit in range(sum(num_qubits)):
    89       200     212514.0   1062.6      0.2             init_dist.ry(noise[num_qubit], num_qubit)
    90
    91       100       4289.0     42.9      0.0         params = cast(np.ndarray, parameter_values)
    92
    93       100   12379363.0 123793.6      9.0         qc.append(construct_circuit(params), quantum)
    94       100    3757671.0  37576.7      2.7         final_circuit = qc.compose(init_dist, front=True)
    95       100     343714.0   3437.1      0.2         final_circuit.measure_all()
    96
    97       100  111187592.0 1111875.9     80.9        final_circuit = qiskit.transpile(final_circuit, simulator)
    98       100    8225711.0  82257.1      6.0         result = simulator.run(final_circuit, shots=shots).result()
    99       100     205308.0   2053.1      0.1         counts = result.get_counts(final_circuit)
   100
   101       100       1549.0     15.5      0.0         try:
   102       100      25741.0    257.4      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      31862.0    318.6      0.0         pixels = pixels / shots
   117       100     618189.0   6181.9      0.4         print(pixels)


Process finished with exit code 0

```