```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/
performance_get_output_generator.py
[0.12792969 0.27880859 0.56005859 0.03320312]
[0.13232422 0.27783203 0.55957031 0.03027344]
[0.1328125  0.24902344 0.58886719 0.02929688]
[0.13378906 0.26904297 0.55859375 0.03857422]
[0.12988281 0.26806641 0.57177734 0.03027344]
[0.15380859 0.26611328 0.55322266 0.02685547]
[0.11767578 0.26513672 0.58105469 0.03613281]
[0.13427734 0.28271484 0.55126953 0.03173828]
[0.13769531 0.29394531 0.53613281 0.03222656]
[0.13476562 0.26855469 0.56494141 0.03173828]
[0.15136719 0.27246094 0.54833984 0.02783203]
[0.13623047 0.26708984 0.56689453 0.02978516]
[0.13818359 0.25732422 0.56787109 0.03662109]
[0.14013672 0.26660156 0.56445312 0.02880859]
[0.12695312 0.28271484 0.55908203 0.03125   ]
[0.13720703 0.28173828 0.54931641 0.03173828]
[0.13574219 0.27539062 0.56347656 0.02539062]
[0.14453125 0.25878906 0.56445312 0.03222656]
[0.12939453 0.24658203 0.59765625 0.02636719]
[0.14599609 0.28808594 0.54052734 0.02539062]
[0.14550781 0.27148438 0.54833984 0.03466797]
[0.13574219 0.27978516 0.55126953 0.03320312]
[0.13232422 0.26708984 0.56933594 0.03125   ]
[0.14648438 0.29638672 0.52978516 0.02734375]
[0.12841797 0.26708984 0.57470703 0.02978516]
[0.13964844 0.25830078 0.57617188 0.02587891]
[0.14013672 0.27978516 0.55224609 0.02783203]
[0.13183594 0.26660156 0.56445312 0.03710938]
[0.14550781 0.26660156 0.5625     0.02539062]
[0.13037109 0.27783203 0.55810547 0.03369141]
[0.14599609 0.28027344 0.54492188 0.02880859]
[0.13916016 0.27197266 0.55908203 0.02978516]
[0.12207031 0.27832031 0.56542969 0.03417969]
[0.13818359 0.28271484 0.54931641 0.02978516]
[0.140625   0.27880859 0.54980469 0.03076172]
[0.13427734 0.25976562 0.57666016 0.02929688]
[0.13134766 0.27490234 0.56542969 0.02832031]
[0.13916016 0.27636719 0.55126953 0.03320312]
[0.14355469 0.26367188 0.56201172 0.03076172]
[0.14306641 0.265625   0.55371094 0.03759766]
[0.13671875 0.27685547 0.55859375 0.02783203]
[0.12597656 0.26855469 0.5703125  0.03515625]
[0.13623047 0.28466797 0.54931641 0.02978516]
[0.13085938 0.26611328 0.57519531 0.02783203]
[0.15039062 0.27001953 0.55078125 0.02880859]
[0.14550781 0.25195312 0.57275391 0.02978516]
[0.12939453 0.28125    0.56591797 0.0234375 ]
[0.11669922 0.27783203 0.57617188 0.02929688]
[0.13818359 0.26464844 0.56445312 0.03271484]
[0.13037109 0.27490234 0.56787109 0.02685547]
[0.14648438 0.27001953 0.54931641 0.03417969]
[0.13916016 0.26513672 0.56005859 0.03564453]
[0.140625   0.27783203 0.56054688 0.02099609]
[0.13574219 0.26855469 0.56787109 0.02783203]
[0.13037109 0.27539062 0.55957031 0.03466797]
[0.12207031 0.27636719 0.56884766 0.03271484]
[0.12548828 0.265625   0.57617188 0.03271484]
[0.15283203 0.25830078 0.55761719 0.03125   ]
[0.13818359 0.26855469 0.55957031 0.03369141]
[0.13623047 0.28417969 0.54394531 0.03564453]
[0.13525391 0.26660156 0.56152344 0.03662109]
[0.14160156 0.27978516 0.55175781 0.02685547]
[0.13525391 0.27246094 0.56152344 0.03076172]
[0.13671875 0.28857422 0.55175781 0.02294922]
[0.14501953 0.27929688 0.54736328 0.02832031]
[0.11914062 0.2734375  0.56884766 0.03857422]
[0.13476562 0.2734375  0.55761719 0.03417969]
[0.13330078 0.26757812 0.56884766 0.03027344]
[0.13134766 0.26806641 0.56298828 0.03759766]
[0.13671875 0.28027344 0.55078125 0.03222656]
[0.13330078 0.28662109 0.54882812 0.03125   ]
[0.13427734 0.26513672 0.56835938 0.03222656]
[0.12744141 0.27490234 0.5703125  0.02734375]
[0.14697266 0.26464844 0.55517578 0.03320312]
[0.13671875 0.27734375 0.55908203 0.02685547]
[0.12988281 0.29541016 0.54296875 0.03173828]
[0.14453125 0.27050781 0.56054688 0.02441406]
[0.13623047 0.27587891 0.55615234 0.03173828]
[0.13330078 0.26318359 0.5703125  0.03320312]
[0.12890625 0.29296875 0.55029297 0.02783203]
[0.13427734 0.26416016 0.57373047 0.02783203]
[0.14550781 0.26855469 0.55273438 0.03320312]
[0.14355469 0.27783203 0.54345703 0.03515625]
[0.140625   0.28125    0.54882812 0.02929688]
[0.13037109 0.27441406 0.55566406 0.03955078]
[0.14355469 0.27197266 0.55712891 0.02734375]
[0.14013672 0.27929688 0.54833984 0.03222656]
```

```
[0.14160156 0.26416016 0.56933594 0.02490234]
[0.14355469 0.26855469 0.55664062 0.03125   ]
[0.15429688 0.25537109 0.55957031 0.03076172]
[0.14208984 0.28564453 0.54443359 0.02783203]
[0.12207031 0.28515625 0.56005859 0.03271484]
[0.1484375  0.26074219 0.56347656 0.02734375]
[0.14257812 0.26269531 0.56298828 0.03173828]
[0.1328125  0.27734375 0.55761719 0.03222656]
[0.13671875 0.26660156 0.56152344 0.03515625]
[0.13085938 0.27246094 0.56640625 0.03027344]
[0.12890625 0.27099609 0.56884766 0.03125   ]
[0.12939453 0.26708984 0.57421875 0.02929688]
[0.13134766 0.27734375 0.56738281 0.02392578]
[0.12841797 0.27197266 0.56640625 0.03320312]
[0.12255859 0.28027344 0.57128906 0.02587891]
[0.13623047 0.28417969 0.55029297 0.02929688]
[0.13671875 0.26220703 0.57080078 0.03027344]
[0.13818359 0.28417969 0.54541016 0.03222656]
[0.14257812 0.26513672 0.56542969 0.02685547]
[0.13867188 0.27539062 0.55517578 0.03076172]
[0.13427734 0.26611328 0.56494141 0.03466797]
[0.12841797 0.27783203 0.56738281 0.02636719]
[0.13232422 0.28857422 0.55029297 0.02880859]
[0.15380859 0.27734375 0.54052734 0.02832031]
[0.12988281 0.27490234 0.56542969 0.02978516]
[0.1328125  0.26660156 0.56445312 0.03613281]
[0.12988281 0.28662109 0.5546875  0.02880859]
[0.12939453 0.27880859 0.55517578 0.03662109]
[0.13232422 0.27587891 0.56738281 0.02441406]
[0.12695312 0.26953125 0.57324219 0.03027344]
[0.12695312 0.27832031 0.56884766 0.02587891]
[0.14355469 0.28808594 0.54150391 0.02685547]
[0.13183594 0.26318359 0.57226562 0.03271484]
[0.11865234 0.28173828 0.56542969 0.03417969]
[0.13818359 0.27246094 0.55957031 0.02978516]
[0.12646484 0.27636719 0.56054688 0.03662109]
[0.13525391 0.25244141 0.58251953 0.02978516]
[0.13769531 0.26904297 0.57128906 0.02197266]
[0.14501953 0.27636719 0.55175781 0.02685547]
[0.14013672 0.25830078 0.57128906 0.03027344]
[0.16162109 0.27490234 0.53320312 0.03027344]
[0.14111328 0.26220703 0.57275391 0.02392578]
[0.13916016 0.26611328 0.56689453 0.02783203]
[0.14697266 0.27099609 0.55517578 0.02685547]
[0.12695312 0.27929688 0.56445312 0.02929688]
[0.13867188 0.26660156 0.56542969 0.02929688]
[0.13525391 0.29931641 0.53320312 0.03222656]
[0.13916016 0.27685547 0.55419922 0.02978516]
[0.14599609 0.28222656 0.54443359 0.02734375]
[0.13867188 0.28271484 0.54150391 0.03710938]
[0.13330078 0.27441406 0.56445312 0.02783203]
[0.13427734 0.26953125 0.56347656 0.03271484]
[0.14355469 0.27099609 0.55517578 0.03027344]
[0.140625   0.27392578 0.54882812 0.03662109]
[0.13720703 0.27636719 0.55175781 0.03466797]
[0.14208984 0.26904297 0.55712891 0.03173828]
[0.12792969 0.27685547 0.56542969 0.02978516]
[0.14599609 0.27001953 0.54931641 0.03466797]
[0.12451172 0.2734375  0.56787109 0.03417969]
[0.13037109 0.27832031 0.56103516 0.03027344]
[0.13964844 0.26660156 0.56445312 0.02929688]
[0.12939453 0.27001953 0.57275391 0.02783203]
[0.13574219 0.27246094 0.55908203 0.03271484]
[0.13720703 0.28662109 0.53955078 0.03662109]
[0.13916016 0.25048828 0.58740234 0.02294922]
[0.14794922 0.26953125 0.54980469 0.03271484]
[0.13867188 0.28417969 0.55419922 0.02294922]
[0.13867188 0.26513672 0.56640625 0.02978516]
[0.1328125  0.25634766 0.57763672 0.03320312]
[0.1328125  0.26708984 0.56933594 0.03076172]
[0.13427734 0.28320312 0.5546875  0.02783203]
[0.13427734 0.27148438 0.56884766 0.02539062]
[0.13964844 0.25732422 0.56542969 0.03759766]
[0.12744141 0.27978516 0.56396484 0.02880859]
[0.14208984 0.26708984 0.56445312 0.02636719]
[0.14306641 0.2578125  0.56933594 0.02978516]
[0.14599609 0.265625   0.55761719 0.03076172]
[0.14160156 0.27099609 0.5546875  0.03271484]
[0.14453125 0.2734375  0.55126953 0.03076172]
[0.13378906 0.28759766 0.54980469 0.02880859]
[0.14697266 0.27197266 0.55273438 0.02832031]
[0.11816406 0.28125    0.56542969 0.03515625]
[0.14306641 0.2578125  0.57177734 0.02734375]
[0.13232422 0.26953125 0.56689453 0.03125   ]
[0.12890625 0.28662109 0.55712891 0.02734375]
[0.14111328 0.26123047 0.56152344 0.03613281]
[0.13769531 0.26416016 0.56787109 0.03027344]
[0.13183594 0.25439453 0.58398438 0.02978516]
[0.13085938 0.28466797 0.55761719 0.02685547]
```

```
[0.12158203 0.27441406 0.57958984 0.02441406]
[0.12597656 0.27783203 0.56787109 0.02832031]
[0.14404297 0.26220703 0.56054688 0.03320312]
[0.12255859 0.28027344 0.56884766 0.02832031]
[0.13330078 0.27587891 0.55712891 0.03369141]
[0.12304688 0.27685547 0.57128906 0.02880859]
[0.14599609 0.25927734 0.56347656 0.03125   ]
[0.13769531 0.26074219 0.56982422 0.03173828]
[0.12695312 0.28662109 0.55029297 0.03613281]
[0.13232422 0.27783203 0.55371094 0.03613281]
[0.13378906 0.27978516 0.55761719 0.02880859]
[0.14355469 0.28027344 0.54541016 0.03076172]
[0.14208984 0.26513672 0.55859375 0.03417969]
[0.14794922 0.25732422 0.56494141 0.02978516]
[0.14599609 0.28417969 0.54101562 0.02880859]
[0.13427734 0.25634766 0.57714844 0.03222656]
[0.13720703 0.25927734 0.57128906 0.03222656]
[0.13720703 0.25634766 0.57763672 0.02880859]
[0.14648438 0.28173828 0.54296875 0.02880859]
[0.13085938 0.27929688 0.55957031 0.03027344]
[0.14746094 0.26171875 0.56494141 0.02587891]
[0.1328125  0.2734375  0.55908203 0.03466797]
[0.12939453 0.27880859 0.56591797 0.02587891]
[0.14453125 0.2734375  0.55126953 0.03076172]
Timer unit: 1e-07 s

Total time: 13.7448 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       3128.0     31.0      0.0      for noise in batch_noise:
    36       100       2755.0     27.6      0.0          real_keys = {"00", "10", "01", "11"}
    37
    38       100      91609.0    916.1      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    39       100     225582.0   2255.8      0.2          qc = QuantumCircuit(sum(num_qubits))
    40
    41       100     150611.0   1506.1      0.1          init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    42       100       3206.0     32.1      0.0          assert noise.shape[0] == sum(num_qubits)
    43
    44       300       4929.0     16.4      0.0          for num_qubit in range(sum(num_qubits)):
    45       200     222818.0   1114.1      0.2              init_dist.ry(noise[num_qubit], num_qubit)
    46
    47       100       4605.0     46.0      0.0          params = cast(np.ndarray, parameter_values)
    48
    49       100   11690774.0 116907.7      8.5          qc.append(construct_circuit(params), quantum)
    50       100    3521568.0  35215.7      2.6          final_circuit = qc.compose(init_dist, front=True)
    51       100     322972.0   3229.7      0.2          final_circuit.measure_all()
    52
    53       100    4176100.0  41761.0      3.0          simulator_1 = qiskit.Aer.get_backend("aer_simulator")
    54       100  107818193.0 1078181.9     78.4          final_circuit = qiskit.transpile(final_circuit, simulator_1)
    55       100    8322257.0  83222.6      6.1          result = simulator_1.run(final_circuit, shots=shots).result()
    56       100     198852.0   1988.5      0.1          counts = result.get_counts(final_circuit)
    57
    58       100       1450.0     14.5      0.0          try:
    59       100      26291.0    262.9      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      33749.0    337.5      0.0          pixels = pixels / shots
    74       100     626704.0   6267.0      0.5          print(pixels)

Total time: 13.0022 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       3011.0   3011.0      0.0      simulator = qiskit.Aer.get_backend("aer_simulator")
    79       101       2839.0     28.1      0.0      for noise in batch_noise:
    80       100       2146.0     21.5      0.0          real_keys = {"00", "10", "01", "11"}
    81
    82       100      88199.0    882.0      0.1          quantum = QuantumRegister(sum(num_qubits), name="q")
    83       100     228570.0   2285.7      0.2          qc = QuantumCircuit(sum(num_qubits))
```

```
    84
    85       100     149883.0   1498.8      0.1         init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    86       100       3251.0     32.5      0.0         assert noise.shape[0] == sum(num_qubits)
    87
    88       300       5155.0     17.2      0.0         for num_qubit in range(sum(num_qubits)):
    89       200     217178.0   1085.9      0.2             init_dist.ry(noise[num_qubit], num_qubit)
    90
    91       100       4126.0     41.3      0.0         params = cast(np.ndarray, parameter_values)
    92
    93       100   11709896.0 117099.0      9.0         qc.append(construct_circuit(params), quantum)
    94       100    3579452.0  35794.5      2.8         final_circuit = qc.compose(init_dist, front=True)
    95       100     354022.0   3540.2      0.3         final_circuit.measure_all()
    96
    97       100  104466391.0 1044663.9     80.3        final_circuit = qiskit.transpile(final_circuit, simulator)
    98       100    8309285.0  83092.9      6.4         result = simulator.run(final_circuit, shots=shots).result()
    99       100     211648.0   2116.5      0.2         counts = result.get_counts(final_circuit)
   100
   101       100       1522.0     15.2      0.0         try:
   102       100      27402.0    274.0      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      37335.0    373.4      0.0         pixels = pixels / shots
   117       100     620323.0   6203.2      0.5         print(pixels)


Process finished with exit code 0

```