A general Tutorial of CktGNN

This tutorial mainly contains three parts:

Section.1 Understand a generated circuit graph from our training framework.
Section. 2. Generate a spice netlist template for each sub-circuit.
Section. 3. Convert a generated circuit graph to a spice netlist for performance simulation.

Section.1 Understand a generated circuit graph from our training framework.

Here, we briefly introduce how the generated circuits (i.e., from graph training) are stored in the txt files. Basically, for each circuit (Fig. 2), the **first line** stores the number of subgraphs (i.e., 7) and the number of all nodes (i.e. 12).

Then the next 7 lines contains information about these subgraphs. In each line (we take line 4 as an example):

1. The first element is the type of the subgraph. Based on the SUBG_NODE and SUBG_CON (Fig. 4), we can determine the subgraph (an example is shown in Fig. 1). For instance, in the line 4, the first element is 11, then it contains a C and a -gm with feedforward (in -gm+, + means feedforward) according to SUBG_NODE, and they are connected in parallel (i.e., Fig. 1).

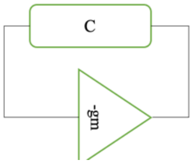
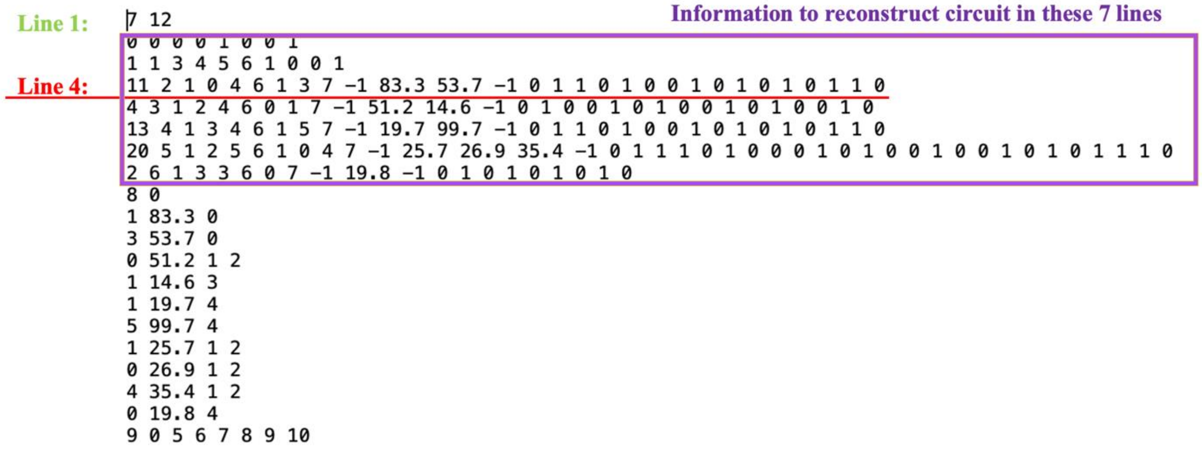


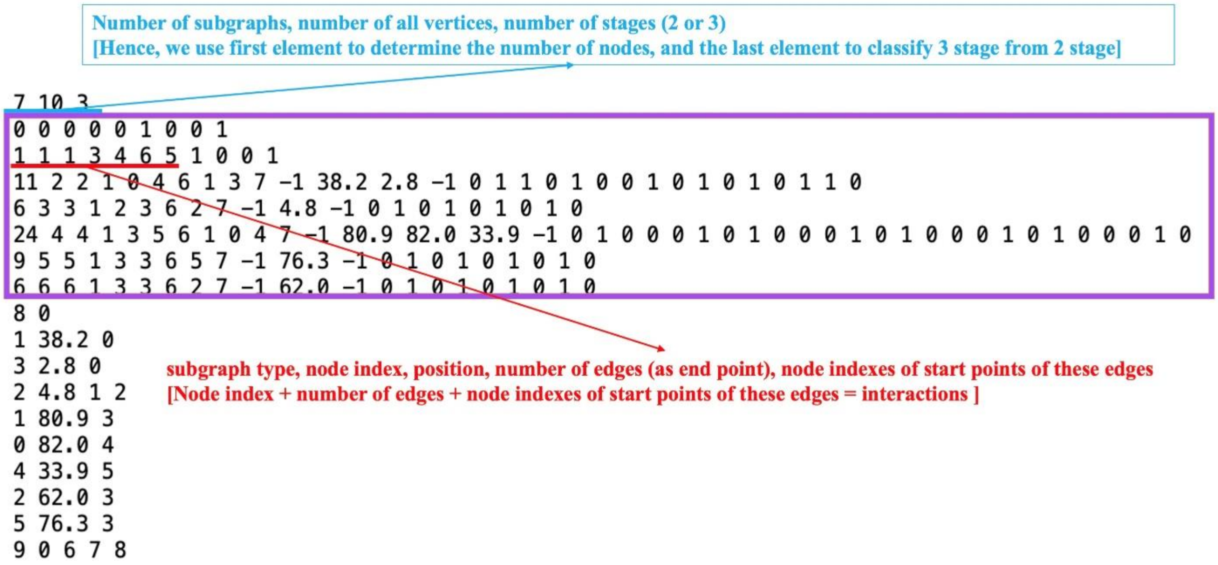
Fig. 1 An example of subgraph. All subgraphs are shown in Fig. 2 in our ICLR 2023 paper.

2. We use directed edges to store the connections between subgraphs. And we can reconstruct whole graph based on these directed edges by following rules. The second element 2 is the id (or node index) in the graph. The third element k=1 is the number of directed edges to the node, and the next k=1 elements are the id of subgraph that there is a directed edge to this subgraph. Then for node 2, we know that the subgraph type is 11, and there is one directed edge from node 0 to node 2. Similarly, if we take the line 3 as example, we find that node 1 takes a subgraph type of 1, there are 3 directed edges to it, which are edges from node 4, node 5, and node 6. Moreover, for simplicity, we add a sudo-edge from GND node (represent the subgraph with largest id, i.e. node 6 in the example of line 3) to the output node (i.e. node 1).
3. Then in the line, we can see that there are some float numbers between -1 and -1 (83.3 and 53.7), these are values for C and gm. We normalize these values to [0,100] for R, C, and gm due to the model training consideration, hence we need to scale them back to the original values when constructing simulation model.



```
7 12
0 0 0 0 1 0 0 1
1 1 3 4 5 6 1 0 0 1
11 2 1 0 4 6 1 3 7 −1 83.3 53.7 −1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0
4 3 1 2 4 6 0 1 7 −1 51.2 14.6 −1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0
13 4 1 3 4 6 1 5 7 −1 19.7 99.7 −1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0
20 5 1 2 5 6 1 0 4 7 −1 25.7 26.9 35.4 −1 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 1 1 0
2 6 1 3 3 6 0 7 −1 19.8 −1 0 1 0 1 0 1 0 1 0
8 0
1 83.3 0
3 53.7 0
0 51.2 1 2
1 14.6 3
1 19.7 4
5 99.7 4
1 25.7 1 2
0 26.9 1 2
4 35.4 1 2
0 19.8 4
9 0 5 6 7 8 9 10
```

Fig. 2 An example of generated circuit graphs.

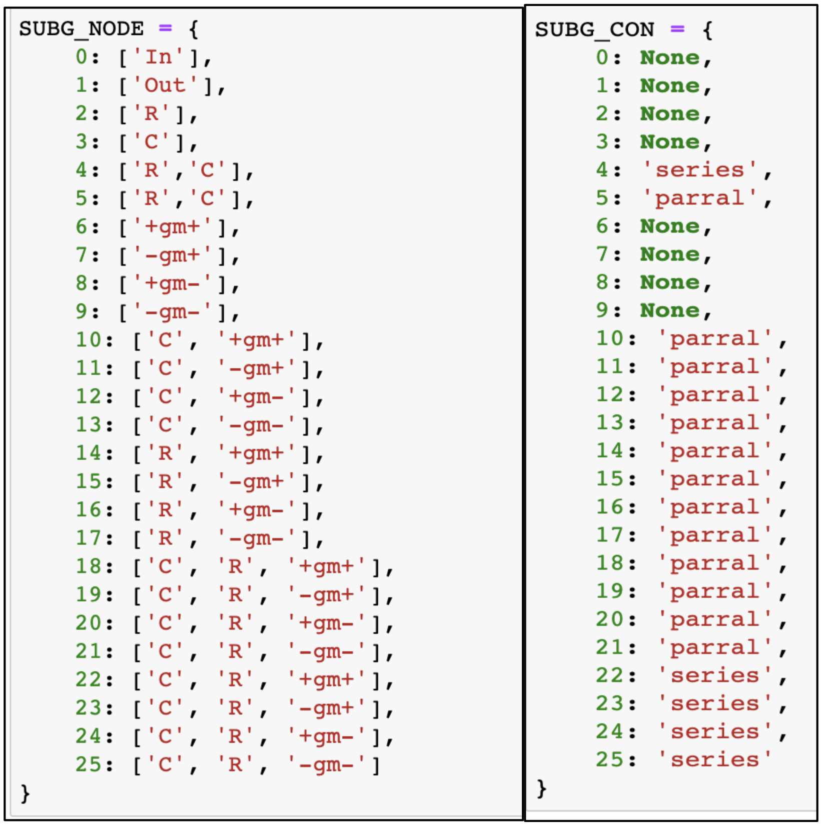


Number of subgraphs, number of all vertices, number of stages (2 or 3)
[Hence, we use first element to determine the number of nodes, and the last element to classify 3 stage from 2 stage]

```
7 10 3
0 0 0 0 0 1 0 0 1
1 1 1 3 4 6 5 1 0 0 1
11 2 2 1 0 4 6 1 3 7 -1 38.2 2.8 -1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 1 0
6 3 3 1 2 3 6 2 7 -1 4.8 -1 0 1 0 1 0 1 0 1 0
24 4 4 1 3 5 6 1 0 4 7 -1 80.9 82.0 33.9 -1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0
9 5 5 1 3 3 6 5 7 -1 76.3 -1 0 1 0 1 0 1 0 1 0
6 6 6 1 3 3 6 2 7 -1 62.0 -1 0 1 0 1 0 1 0 1 0
8 0
1 38.2 0
3 2.8 0
2 4.8 1 2
1 80.9 3
0 82.0 4
4 33.9 5
2 62.0 3
5 76.3 3
9 0 6 7 8
```

subgraph type, node index, position, number of edges (as end point), node indexes of start points of these edges
[Node index + number of edges + node indexes of start points of these edges = interactions]

Fig. 3 Another example of generated circuit graphs.



```
SUBG_NODE = {
    0: ['In'],
    1: ['Out'],
    2: ['R'],
    3: ['C'],
    4: ['R','C'],
    5: ['R','C'],
    6: ['+gm+'],
    7: ['-gm+'],
    8: ['+gm-'],
    9: ['-gm-'],
    10: ['C', '+gm+'],
    11: ['C', '-gm+'],
    12: ['C', '+gm-'],
    13: ['C', '-gm-'],
    14: ['R', '+gm+'],
    15: ['R', '-gm+'],
    16: ['R', '+gm-'],
    17: ['R', '-gm-'],
    18: ['C', 'R', '+gm+'],
    19: ['C', 'R', '-gm+'],
    20: ['C', 'R', '+gm-'],
    21: ['C', 'R', '-gm-'],
    22: ['C', 'R', '+gm+'],
    23: ['C', 'R', '-gm+'],
    24: ['C', 'R', '+gm-'],
    25: ['C', 'R', '-gm-']
}
```

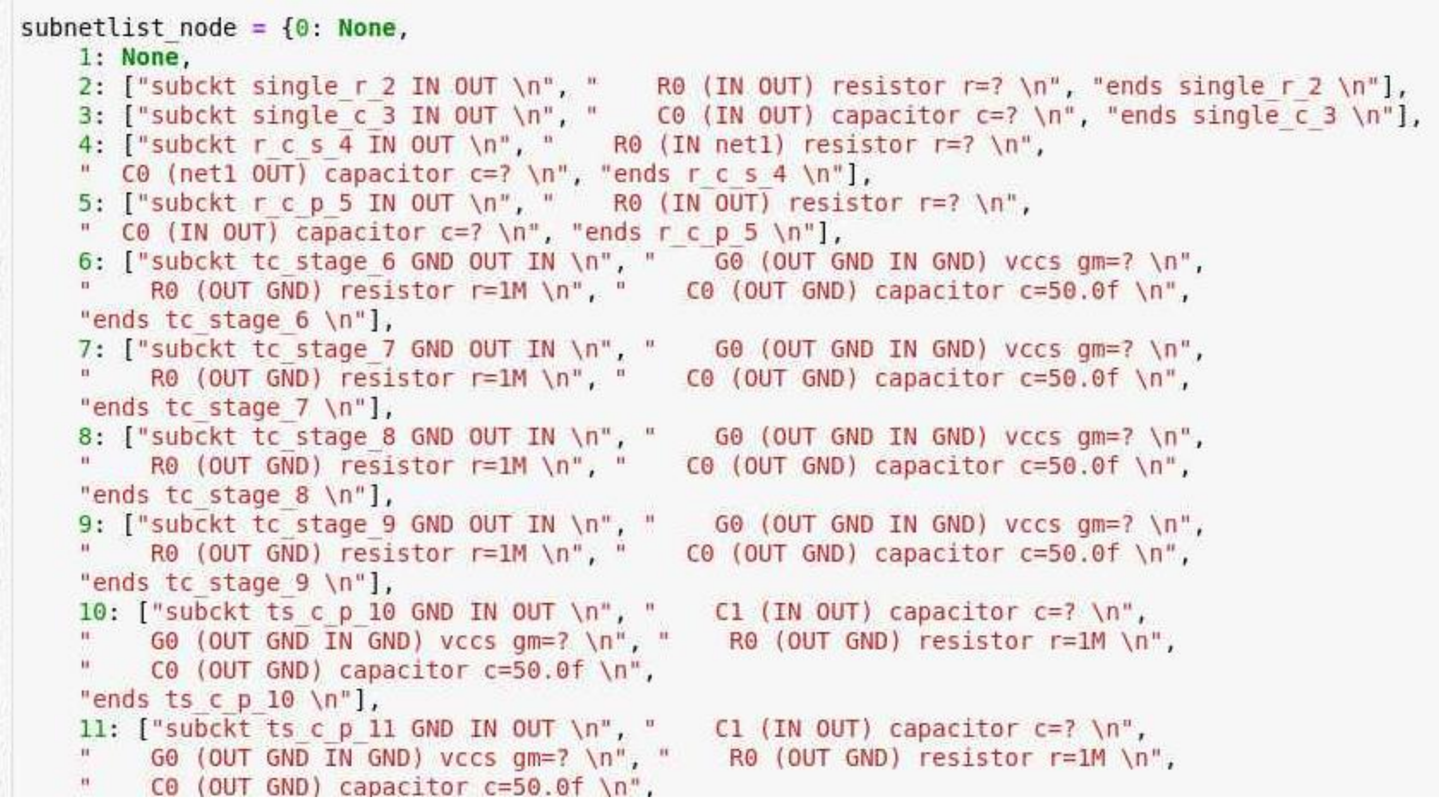
```
SUBG_CON = {
    0: None,
    1: None,
    2: None,
    3: None,
    4: 'series',
    5: 'parral',
    6: None,
    7: None,
    8: None,
    9: None,
    10: 'parral',
    11: 'parral',
    12: 'parral',
    13: 'parral',
    14: 'parral',
    15: 'parral',
    16: 'parral',
    17: 'parral',
    18: 'parral',
    19: 'parral',
    20: 'parral',
    21: 'parral',
    22: 'series',
    23: 'series',
    24: 'series',
    25: 'series'
}
```

Fig. 4 Encoding nodes of a general op-amp using python.

Section. 2. Generate a spice netlist template for each sub-circuit.

For each sub-graph listed in Fig. 4, we have created a **spice netlist template** for it. Note that **due to IP issues, we only use the behavioral model of each sub-circuit to generate its spice netlist in our open-sourced packages**. For example, for each trans-conductance (tc) stage (gm), we leverage a voltage-controlled-current-source to replace it. Below we show some examples.

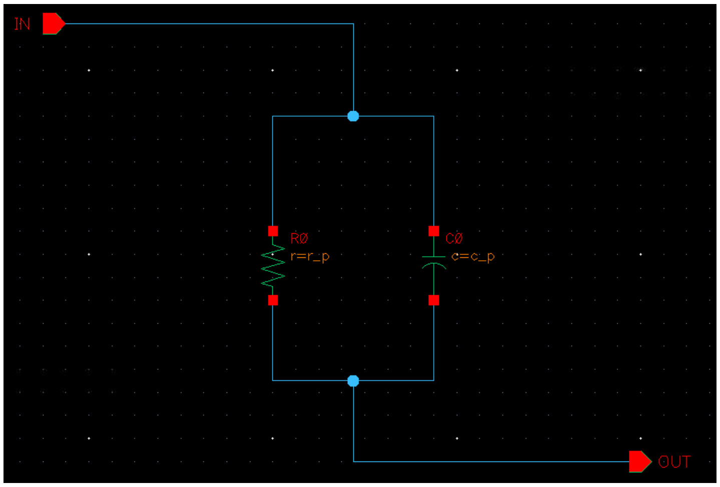
We also provide a python script in **Section. 3** that can convert an arbitrary circuit graph from our training framework into a spice netlist for simulation. Please look at the attached file for a practice. It is very easy to use this script by looking at its inside. Therefore, details are omitted. But feel free to contact us if there are any questions.



```
subnetlist_node = {0: None,
    1: None,
    2: ["subckt single_r_2 IN OUT \n", "    R0 (IN OUT) resistor r=? \n", "ends single_r_2 \n"],
    3: ["subckt single_c_3 IN OUT \n", "    C0 (IN OUT) capacitor c=? \n", "ends single_c_3 \n"],
    4: ["subckt r_c_s_4 IN OUT \n", "    R0 (IN net1) resistor r=? \n",
    "  C0 (net1 OUT) capacitor c=? \n", "ends r_c_s_4 \n"],
    5: ["subckt r_c_p_5 IN OUT \n", "    R0 (IN OUT) resistor r=? \n",
    "  C0 (IN OUT) capacitor c=? \n", "ends r_c_p_5 \n"],
    6: ["subckt tc_stage_6 GND OUT IN \n", "    G0 (OUT GND IN GND) vccs gm=? \n",
    "    R0 (OUT GND) resistor r=1M \n", "    C0 (OUT GND) capacitor c=50.0f \n",
    "ends tc_stage_6 \n"],
    7: ["subckt tc_stage_7 GND OUT IN \n", "    G0 (OUT GND IN GND) vccs gm=? \n",
    "    R0 (OUT GND) resistor r=1M \n", "    C0 (OUT GND) capacitor c=50.0f \n",
    "ends tc_stage_7 \n"],
    8: ["subckt tc_stage_8 GND OUT IN \n", "    G0 (OUT GND IN GND) vccs gm=? \n",
    "    R0 (OUT GND) resistor r=1M \n", "    C0 (OUT GND) capacitor c=50.0f \n",
    "ends tc_stage_8 \n"],
    9: ["subckt tc_stage_9 GND OUT IN \n", "    G0 (OUT GND IN GND) vccs gm=? \n",
    "    R0 (OUT GND) resistor r=1M \n", "    C0 (OUT GND) capacitor c=50.0f \n",
    "ends tc_stage_9 \n"],
    10: ["subckt ts_c_p_10 GND IN OUT \n", "    C1 (IN OUT) capacitor c=? \n",
    "    G0 (OUT GND IN GND) vccs gm=? \n", "    R0 (OUT GND) resistor r=1M \n",
    "    C0 (OUT GND) capacitor c=50.0f \n",
    "ends ts_c_p_10 \n"],
    11: ["subckt ts_c_p_11 GND IN OUT \n", "    C1 (IN OUT) capacitor c=? \n",
    "    G0 (OUT GND IN GND) vccs gm=? \n", "    R0 (OUT GND) resistor r=1M \n",
    "    C0 (OUT GND) capacitor c=50.0f \n",
// Library name: GNN_Circuit
// Cell name: tc-stage
// View name: schematic
subckt _sub0 GND OUT VIN
    R0 (OUT GND) resistor r=r
    C0 (OUT GND) capacitor c=c
    G0 (OUT GND VIN GND) vccs gm=gm
ends _sub0
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: ab_op_amp
// View name: schematic
I1 (0 VOUT net1) _sub0
I0 (0 net1 net2) _sub0
V0 (net2 0) vsource type=sine ampl=1m sinephase=0 freq=1K
```

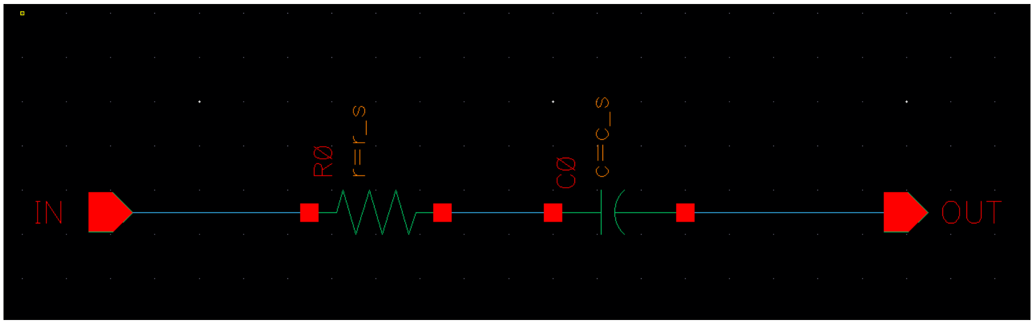
r_c_p

```
// Generated for: spectre
// Generated on: Sep 11 13:02:39 2022
// Design library name: GNN_Circuit
// Design cell name: r_c_p_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters c_p=1p r_p=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: r_c_p
// View name: schematic
subckt r_c_p IN OUT
    R0 (IN OUT) resistor r=r_p
    C0 (IN OUT) capacitor c=c_p
ends r_c_p
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: r_c_p_test
// View name: schematic
I0 (net1 OUT) r_c_p
V0 (net1 0) vsource type=dc
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



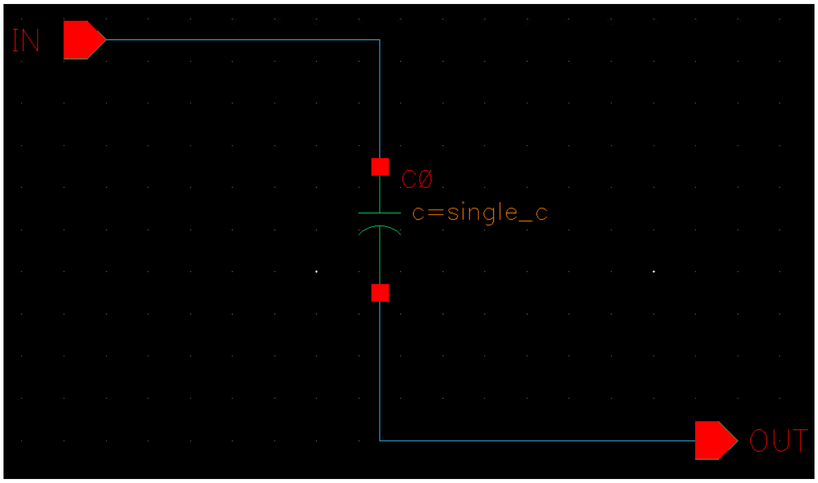
r_c_s

```
// Generated for: spectre
// Generated on: Sep 11 13:12:01 2022
// Design library name: GNN_Circuit
// Design cell name: r_c_s_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters c_s=1p r_s=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: r_c_s
// View name: schematic
subckt r_c_s IN OUT
    C0 (net1 OUT) capacitor c=c_s
    R0 (IN net1) resistor r=r_s
ends r_c_s
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: r_c_s_test
// View name: schematic
I0 (net1 OUT) r_c_s
V0 (net1 0) vsource dc=1 type=dc
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



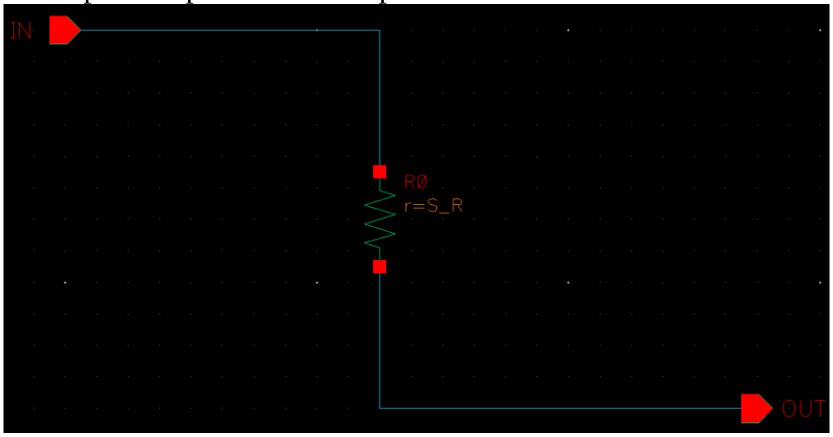
single_c

```
// Generated for: spectre
// Generated on: Sep 11 13:24:09 2022
// Design library name: GNN_Circuit
// Design cell name: single_c_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters single_c=1p wireopt=211

// Library name: GNN_Circuit
// Cell name: single_c
// View name: schematic
subckt single_c IN OUT
    C0 (IN OUT) capacitor c=single_c
ends single_c
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: single_c_test
// View name: schematic
I0 (net1 OUT) single_c
V0 (net1 0) vsource dc=1 type=dc
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



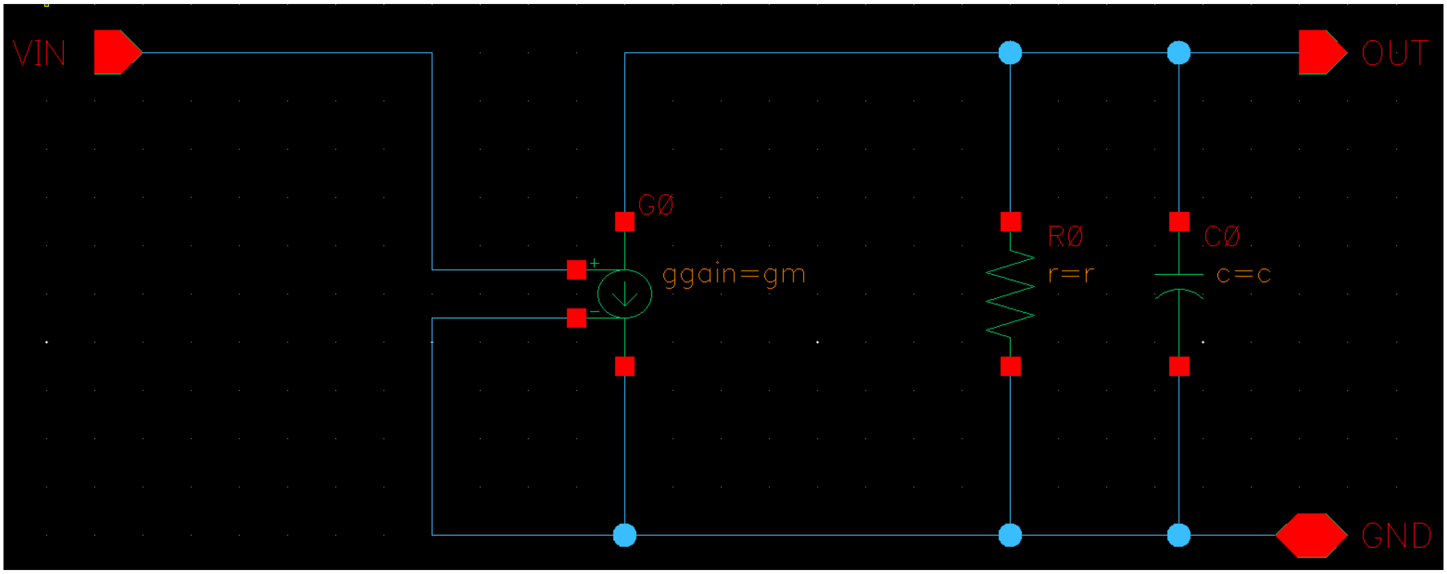
single_r

```
// Generated for: spectre
// Generated on: Sep 11 13:27:52 2022
// Design library name: GNN_Circuit
// Design cell name: single_r_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters S_R=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: single_r
// View name: schematic
subckt single_r IN OUT
    R0 (IN OUT) resistor r=S_R
ends single_r
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: single_r_test
// View name: schematic
V0 (net1 0) vsource type=dc
I1 (net1 OUT) single_r
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



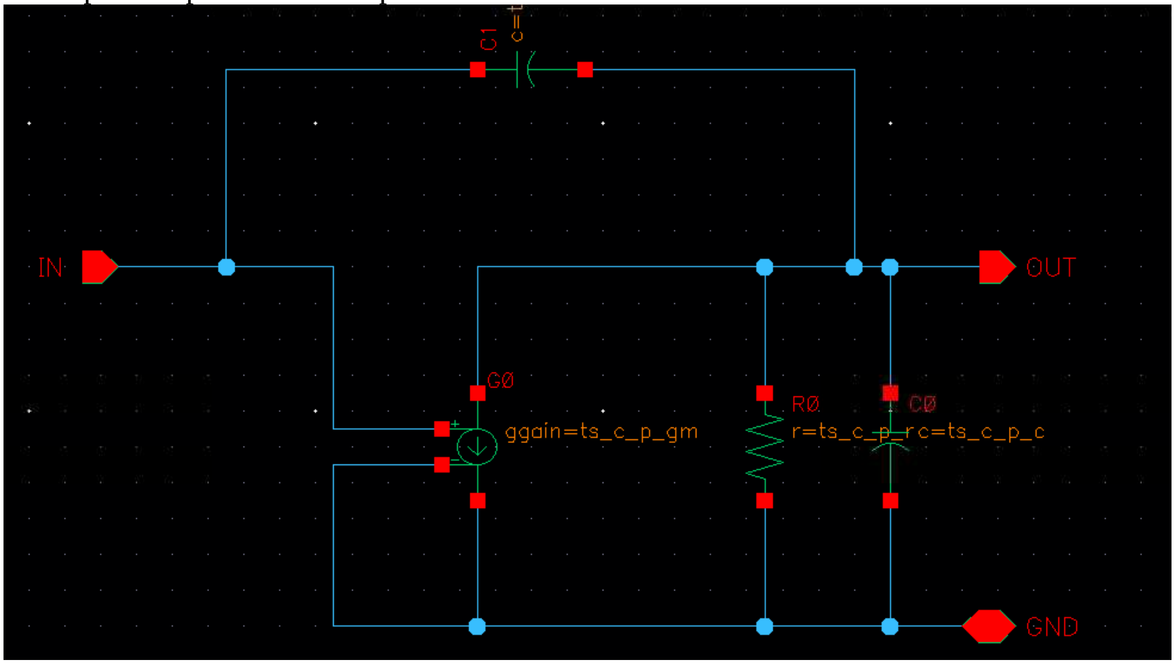
tc_stage

```
// Generated for: spectre
// Generated on: Sep 11 13:37:01 2022
// Design library name: GNN_Circuit
// Design cell name: tc_stage
// Design view name: schematic
simulator lang=spectre
global 0
parameters c=1p gm=1m r=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: tc_stage
// View name: schematic
R0 (OUT GND) resistor r=r
C0 (OUT GND) capacitor c=c
G0 (OUT GND VIN GND) vccs gm=gm
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



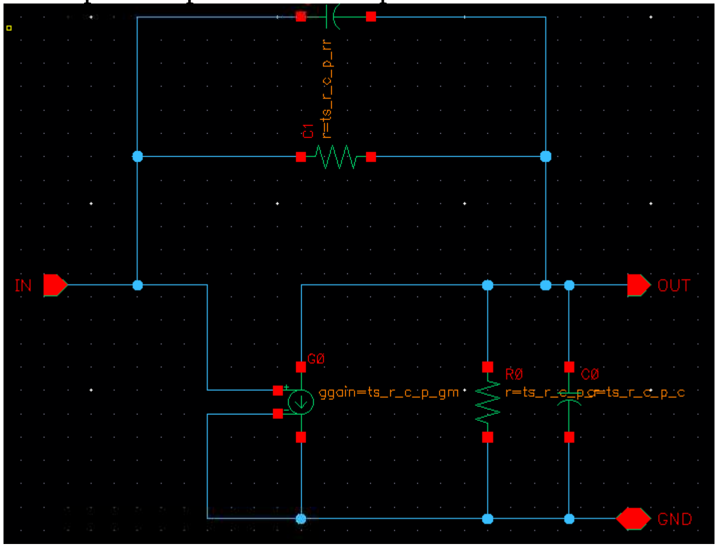
ts_c_p

```
// Generated for: spectre
// Generated on: Sep 11 13:53:03 2022
// Design library name: GNN_Circuit
// Design cell name: ts_c_p
// Design view name: schematic
simulator lang=spectre
global 0
parameters ts_c_p_c=1p ts_c_p_gm=1m ts_c_p_gm_cc=1p ts_c_p_r=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: ts_c_p
// View name: schematic
R0 (OUT GND) resistor r=ts_c_p_r
C1 (IN OUT) capacitor c=ts_c_p_gm_cc
C0 (OUT GND) capacitor c=ts_c_p_c
G0 (OUT GND IN GND) vccs gm=ts_c_p_gm
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



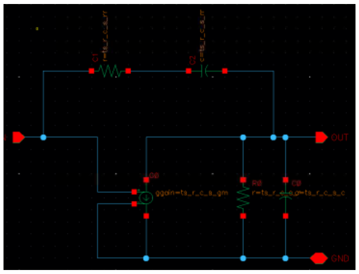
ts_r_c_p

```
// Generated for: spectre
// Generated on: Sep 11 13:56:59 2022
// Design library name: GNN_Circuit
// Design cell name: ts_r_c_p
// Design view name: schematic
simulator lang=spectre
global 0
parameters ts_r_c_p_c=1p ts_r_c_p_cc=1p ts_r_c_p_gm=1m ts_r_c_p_r=1M \
    ts_r_c_p_rr=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: ts_r_c_p
// View name: schematic
C1 (IN OUT) resistor r=ts_r_c_p_rr
R0 (OUT GND) resistor r=ts_r_c_p_r
C2 (IN OUT) capacitor c=ts_r_c_p_cc
C0 (OUT GND) capacitor c=ts_r_c_p_c
G0 (OUT GND IN GND) vccs gm=ts_r_c_p_gm
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



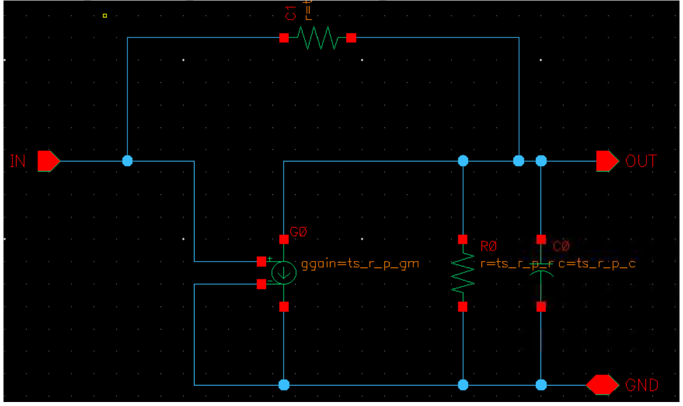
ts_r_c_s

```
// Generated for: spectre
// Generated on: Sep 11 14:24:04 2022
// Design library name: GNN_Circuit
// Design cell name: ts_r_c_s_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters ts_r_c_s_c=1p ts_r_c_s_gm=1m ts_r_c_s_r=1M ts_r_c_s_rr=1M \
    wireopt=211

// Library name: GNN_Circuit
// Cell name: ts_r_c_s
// View name: schematic
subckt ts_r_c_s GND IN OUT
    C1 (IN net10) resistor r=ts_r_c_s_rr
    R0 (OUT GND) resistor r=ts_r_c_s_r
    C2 (net10 OUT) capacitor c=ts_r_c_s_rr
    C0 (OUT GND) capacitor c=ts_r_c_s_c
    G0 (OUT GND IN GND) vccs gm=ts_r_c_s_gm
ends ts_r_c_s
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: ts_r_c_s_test
// View name: schematic
V0 (net1 0) vsource dc=1 type=dc
I1 (0 net1 OUT) ts_r_c_s
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



ts_r_p

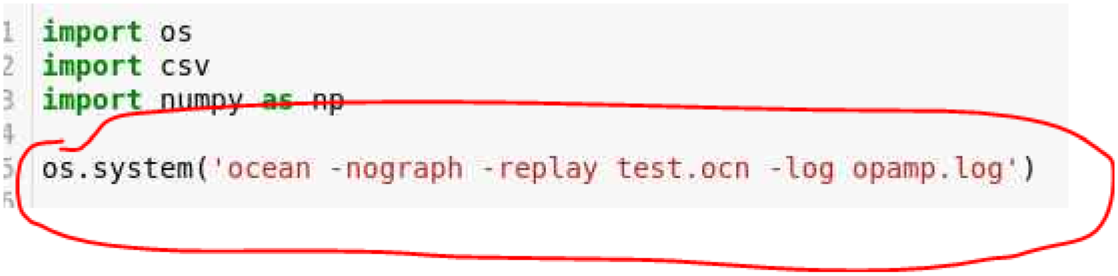
```
// Generated for: spectre
// Generated on: Sep 11 14:28:52 2022
// Design library name: GNN_Circuit
// Design cell name: ts_r_p_test
// Design view name: schematic
simulator lang=spectre
global 0
parameters ts_r_p_c=1p ts_r_p_gm=1m ts_r_p_r=1M ts_r_p_rr=1M wireopt=211

// Library name: GNN_Circuit
// Cell name: ts_r_p
// View name: schematic
subckt ts_r_p GND IN OUT
    C1 (IN OUT) resistor r=ts_r_p_rr c=ts_r_p_rr
    R0 (OUT GND) resistor r=ts_r_p_r
    C0 (OUT GND) capacitor c=ts_r_p_c
    G0 (OUT GND IN GND) vccs gm=ts_r_p_gm
ends ts_r_p
// End of subcircuit definition.

// Library name: GNN_Circuit
// Cell name: ts_r_p_test
// View name: schematic
I0 (0 net1 OUT) ts_r_p
V0 (net1 0) vsource dc=1 type=dc
simulatorOptions options psfversion="1.4.0" reltol=1e-3 vabstol=1e-6 \
    iabstol=1e-12 temp=27 tnom=27 scalem=1.0 scale=1.0 gmin=1e-12 rforce=1 \
    maxnotes=5 maxwarns=5 digits=5 cols=80 pivrel=1e-3 \
    sensfile="../psf/sens.output" checklimitdest=psf
dcOp dc write="spectre.dc" maxiters=150 maxsteps=10000 annotate=status
dcOpInfo info what=oppoint where=rawfile
modelParameter info what=models where=rawfile
element info what=inst where=rawfile
outputParameter info what=output where=rawfile
designParamVals info what=parameters where=rawfile
primitives info what=primitives where=rawfile
subckts info what=subckts where=rawfile
saveOptions options save=allpub
```



Section. 3. Convert a generated circuit graph to a spice netlist for performance simulation.

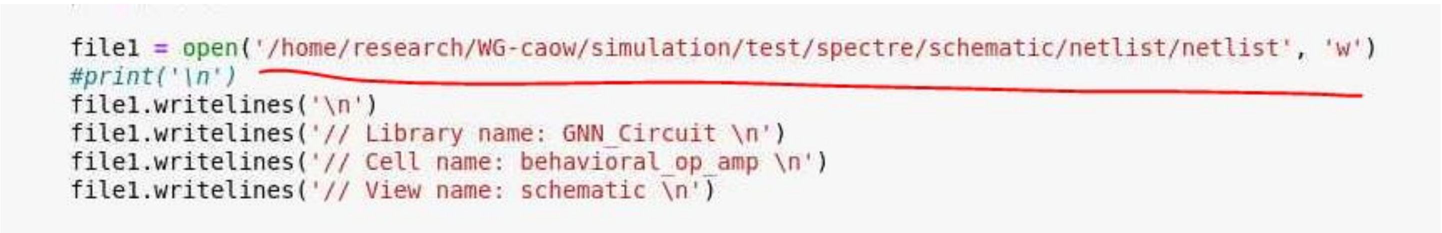
After training the GNN, the last step is to convert a generated circuit graph to a spice netlist for performance simulation. We have developed a python script which can trigger Cadence for simulation through command line. Several key points to pay attention here.

1. Command to trigger Cadence for simulation without using GUI is:



```
import os
import csv
import numpy as np

os.system('ocean -nograph -replay test.ocn -log opamp.log')
```

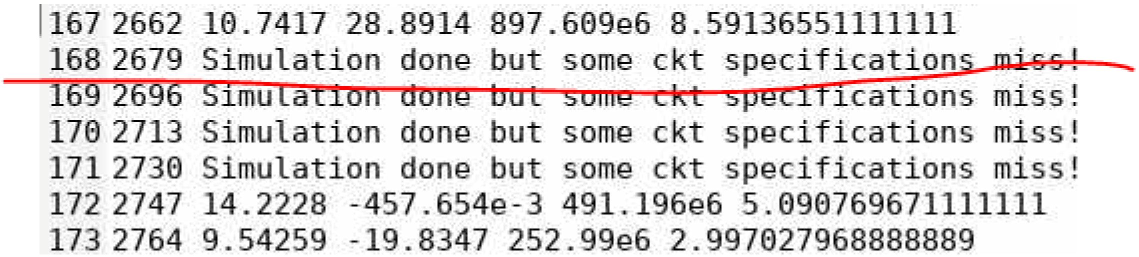
2. To use the above command to simulate a circuit, you first need to create a local folder to store the generated circuit netlist in your Linux environment where Cadence is installed. On our side, an example is shown below:



```
file1 = open('/home/research/WG-caow/simulation/test/spectre/schematic/netlist/netlist', 'w')
#print('\n')
file1.writelines('\n')
file1.writelines('// Library name: GNN_Circuit \n')
file1.writelines('// Cell name: behavioral_op_amp \n')
file1.writelines('// View name: schematic \n')
```

3. Then you can refer to our provided Python script (i.e., gnn_ckt_spec_0) to for simulation.

4. We provide 100K Op-Amp circuit graphs generated from our training framework and their associated performance metrics. Not that not all circuit graphs are meaningful for simulation. For those ill circuit topologies generated by our framework, their simulation results are labelled as



```
167 2662 10.7417 28.8914 897.609e6 8.59136551111111
168 2679 Simulation done but some ckt specifications miss!
169 2696 Simulation done but some ckt specifications miss!
170 2713 Simulation done but some ckt specifications miss!
171 2730 Simulation done but some ckt specifications miss!
172 2747 14.2228 -457.654e-3 491.196e6 5.09076967111111
173 2764 9.54259 -19.8347 252.99e6 2.997027968888889
```