Problem C: Static IR Drop Estimation Using Machine Learning

Gana Surya Prakash Kadagala and Vidya A. Chhabria
Arizona State University
Supported by: Steel Perlot and the OpenROAD Project

Q&A

Q1. I would like to ask topic chair Dr.Chhabria about problem C.
On page 4, spice-based data is provided.
Consider the following netlists,
R34269 n1_1080_1080_2 n1_600_1080_5 10.000000
R36241 n1_167996_60_7 n1_167460_60_8 10.000000

Is it possible to have a resistive component with different layers(via) and different xy locations(wire) at the same time?
Thank you.
A1. No, this will not be possible. If the layer changes, the x and y will be the same.

Q2. I would like to ask about problem C.
The website *https://github.com/UMN-EDA/BeGAN-benchmarks* has provided SPICE-based data. However, there are no matrices about effective distance and PDN density.
Could you please help to advise where the Image-based data provided? Thank you.
A2. These matrices will be released on the website and will be updated on the GitHub repo soon.

Q3. I would like to ask about problem C.
There is something strange in the spice files provided on the website *https://github.com/UMN-EDA/BeGAN-benchmarks*. For example, in real-circuit-enchmarks/asap7/data/aes_reg_grid.sp.gz, the layer changes from M7 to M8, and the x also changes.
R35161 n1_M7_4156_60 n1_M8_4380_60 10.000000
The format of the node in the netlist is also different from that of the document.
<netname>_<x-cordinate>_<y-cordinate>_<layer-idx>
n1_M7_4156_60

What's more, could you please help to advise when the Image-based data will be provided?
Thank you.
A3. This will be fixed shortly and the updated image-based data will be available by May 31st.

Q4. I have a few questions hoping to get clarifications:
1) Are we supposed to use the data provided in https://github.com/UMN-EDA/BeGAN-benchmarks as training data?
2) The problem description states that for the image data, three CSV files (current_map.csv, effective_distance.csv, and pdn_density.csv) are provided, and we are asked to predict an outputvoltage_map.csv. However, in the aforementioned GitHub repo, the data is not provided in this format. See the attached screenshot for details. Specifically, for the BeGAN_000 samples, four png files are provided. I assume they should correspond to the three input CSV files, and one target output voltage map file. However, based on the png file name, I couldn't tell which one corresponds to the pdn_density.csv, and which one corresponds to effective_distance.csv; moreover, I am confused what the purposes cdf.png and Hist.png are in the screenshot
3) I noticed that for Problem A and B, a link to the test cases are provided in the official ICCAD contest webpage, but Problem C doesn't. I wonder where the test case is. According to the contest timeline, the Alph test submission June 19th is fast approaching.
4) Does computing resources be provided? I see it says coming shortly.
Thanks for your time and help. Look forward to your reply.
A4. Here are the answers for these questions.
1) The benchmarks will be released on another website on or before May 31.
2) Yes these will be released with the benchmarks on May 31.
3) Yes we might consider updating this alpha submission for Problem C. Thank you so much for your patience.
4) We are working on this. But this will mostly be provided after the alpha submission because we do not have compute resources for all the teams currently.

Q5. We have a few concerns regarding Problem C and would greatly appreciate your assistance in clarifying them.
Upon reviewing the image-based data provided in the two Cases for Problem C, we observed that the size of each image data is different. We are wondering if this discrepancy arises from the fact that the corresponding PDN is not the entire chip but rather a specific section of the chip. Alternatively, could it be due to variations in the specifications of each chip itself? Clarifying this will help us better understand the problem requirements and tailor our approach accordingly.
A5. Each datapoint can be of a different size. This is due to the fact that that chip area is different. In a single datapoint all images should be of the same size (if you find instances where they are not please bring it to my attention.) However, across multiple data points, this can change. For instance, a chip can be 200X200um and all its voltage drop maps, current maps, PDN maps, and effective distance maps will be 200x200 sized matrices. Another chip can be 500umX500um in area and so their matrices will be 500x500. Your ML model has to be independent of the area of the chip.

Q6. We saw that there are a hundred fake circuits. Are we only allowed to train our model using this dataset? Or can we generate data of ourselves as described in the reference paper of BeGAN?
And we wonder how the PDN map is calculated. Thanks.
A6. You are welcome to generate data as you wish to train the model if you wish. That can be your contributions and approach to the problem. Ultimately the contest requires good score for the secret test benchmarks. The approach you take to train your model is upto you.
The PDN map is synthesized for different regions randomly using a template-like approach. The different templates vary in their PDN densities. For instance, template 1 has a different PDN pitch than template 2. If you want the exact value of PDN pitches that were used, you can extract this from the .sp file.

Q7. I would like to ask the topic chair about problem C.
For Problem C, there are two options to use to train our model; image and data training sets. Currently, the GitHub repo that is supposed to be used for this task, https://github.com/UMN-EDA/BeGAN-benchmarks, is not complete as mentioned by the previous contestants' questions. And it was also mentioned that a new link to be used as an input dataset will be announced on May 31st but we didn't get an update on that too. Is there any other way that the topic chairs are thinking of using as input datasets or will there be an update on the existing datasets? Thank you.
A7. The new link is posted on the ICCAD website. http://iccad-contest.org/Problems.html. It has the complete dataset and benchmarks. Fake and real circuit datasets. It was posted on 6/3. These are the benchmarks to be used. NOT the BeGAN benchmarks.

Q8. In the providing testcased (real circuit / fake circuit) there are 5 types of files
current_map.csv
eff_dist_map.csv
ir_drop_map.csv
pdn_density.csv
netlist.sp
1) I am curious about how to transform .sp files to .csv format. We have looked into the .sp files and tried to measure the node numbers, but it seems they are not the same.
For example :
in fake-circuit-data/current_map00_current.csv the dimension is 821*820.However, the last node(resistant component) from fake-circuit-data/current_map00.sp locate at (1636800, 0) (1641600, 0)is
R391 n1_m1_1636800_0 n1_m1_1641600_0 5.356235
Could you show how to transform this resistance component to .csv?
2) Could we get the mapping function (files) that map .sp to .csv ?
Thanks for your help. Please let me know if I misunderstand the spec.
A8. Here are the answers for the questions.
1) The locations in the spice file can be converted to the csv file by using a 2000dbu. For instance, 113400 is the .sp file location 113400/2000 = 56.7 um. The matrix in the .csv files is at a 1um granularity. So in the above example,

the 821x820 has come by dividing the node in the resistance you highlighted by 2000. I have updated the ccontest document to reflect this.

2) You have understood the spec. I only have code that converts the .sp file to .csv. Not the other way around. This code is simple enough for participants to create.

Q9. In the SPICE-based dataset, how can we obtain the voltage of all nodes? Currently, we only have a few node voltages available from the SPICE-based data. I assume we should acquire the label data (node voltages) or calculate the data ourselves?

A9. The label is the same as the IR drop.csv matrix. Ultimately, even if you use the .sp file the output should be an IR drop.csv matrix. So if you predict the IR drop on a per-node basis (for example what you would get after running SPICE), you would need to convert this to a .csv file. This is just to simplify it for the organizers.

Q10. I would like to ask about problem C.

There may be something wrong in the data provided on June 3rd. The IR drop numbers in fake-circuit-data/map12 map18 map62 map96 and real-circuit-data/ testcase11 testcase12 are less than 0 or greater than the supply voltage 1.1 V.

Thank you.

A10. Thanks for pointing this out. This is a bug. I just fixed it and updated with the 6/13 document and benchmarks.

Q11. I have some questions about problem C.

1) Is the shape of all the images in the dataset square? For example, are they all 420*420 pixels, or are there some images that are rectangular with different lengths and widths?

2) Why are the voltage and current's dimensions not same in the spice files provided on the website https://github.com/UMN-EDA/BeGAN/benchmarks ? Under the condition that every pixel represents the current or voltage in a specified size region, the feature and target dimensions should match.

For example, in BeGAN-benchmarks/BeGAN-circuit-benchmarks/asap7/data, BeGAN_000 had 74 lines and 74 columns in its current data, but had only 71 lines and 71 columns in its voltage data. In BeGAN - benchmarks/BeGAN - circuit - benchmarks/nangate45 / characters/data，BeGAN_001 had 601 lines and 601 columns in its current data, but had only 590 lines and 590 columns in its voltage data

I would appreciate it if you could clarify this for me. Thank you for your time and attention.

A11. Here are the answers for the questions.

1) Not necesarrily square. A chip can be 500um X 600um. Then the .csv file will be 500x600 in size.

2) Please use the benchmarks on the ICCAD website where this problem is fixed. This is also updated in the 6/13 version. It is also mentioned on the BeGAN-benchmakrs github link.

Q12. 1. I have some questions about problem C.Is there a relationship between fake and real data ? Is fake data generated from real data?Is it possible that we can generate more fake data with the data provided, do you support this approach?

A12. The difference between fake and real circuit data is that the current maps in the fake circuits have been generated by generative adversarial networks as described in https://ieeexplore.ieee.org/document/9643566. The PDNs have been assigned to each region just as we do in the real circuit and the voltage source locations are also assigned in the same way as we did for real circuits. The same IR drop analysis tool is used to generate the golden labels. So the relationship between inputs and outputs in both cases is the same.

Q13. The problem description suggests that we should use Python 3.7 for development. However, after investigation, we found that the tsri machine only offers Python 3.6 and 2.7.
According to the support chart, we can still use up to tensorflow==2.6 or torch==1.9.1 if we are using Python 3.6, which is somehow old.
Furthermore, it is mentioned that the BeGAN-benchmarks should not be included. In light of this, could you please provide us with more training data sets? The current training set only contains 100 files, which is significantly smaller compared to the 5000 datapoints mentioned in the paper IREDGe. Thank you.

A13. For now, working with Python 3.6 on the TSRI machine is okay. I am working with the contest organizers as well to figure out what is TSRI machine can support a later version of Python.
I will release about 1000 more fake benchmarks by 6/16.

Q14. Is it normal to have a negative IR drop value in some fabricated data?
For example, the current_map_12_ir_drop_map.csv file contains a value of -5.5e+04, indicating that a certain power grid node gains more voltage than VDD.
I have observed that real circuit data does not exhibit such ambiguous negative IR drop in any testcase ir_drop_map.csv.

A14. This has been fixed with the updated benchmarks on Jun 13th.
I hope this helps! Good luck.

Q15. We are curious about the unit of the coordinate in the SPICE file. For example, in testcase1, the area of the circuit is 298x298 um^2, which is specified by the "[map].csv" files. But there are multiple spice nets located at positions with coordinates larger than 298000 nm. This situation is quite unreasonable so we are confused which is the actual chip size. Moreover, the problem description document used an example of a chip with 80x80um size but there is a via component mentioned in spice netlist located at 113400_179200 in nanometer scale which is definitely out of the aforementioned chip area.

A15. In the updated ICCAD contest document posted on the webiste on 6/12. This has been fixed. The SPICE units is 2000 = 1um. Its the same concept as dbu in a DEF file. I hope this helps.

Q16. In the IR drop map, there are some data points that indicate weird IR drop, such as an area in testcase11 and testcase12 where the IR drop in this place goes up to 0.55V, some data points with 100+kV and -4kV IR drop in fake circuit data 12, 18, 62, 96, and multiple points with negative IR drop. We are confused about whether this value is normal in functional IC.

A16. This has been fixed as of 6/13 as well.

Q17. In the problem description, the inference time is limited to 2 seconds. We would like to know how can we make sure our model can run for less than 2 seconds per case on your device.

A17. This runtime constraint is primarily for inference. It not necessarily fr reading and parsing data. I will shortly update more details on runtime and evaluation metrics.

Q18. In following cases

fake-circuit-data/current_map12_ir_drop_map.csv

fake-circuit-data/current_map18_ir_drop_map.csv

fake-circuit-data/current_map62_ir_drop_map.csv

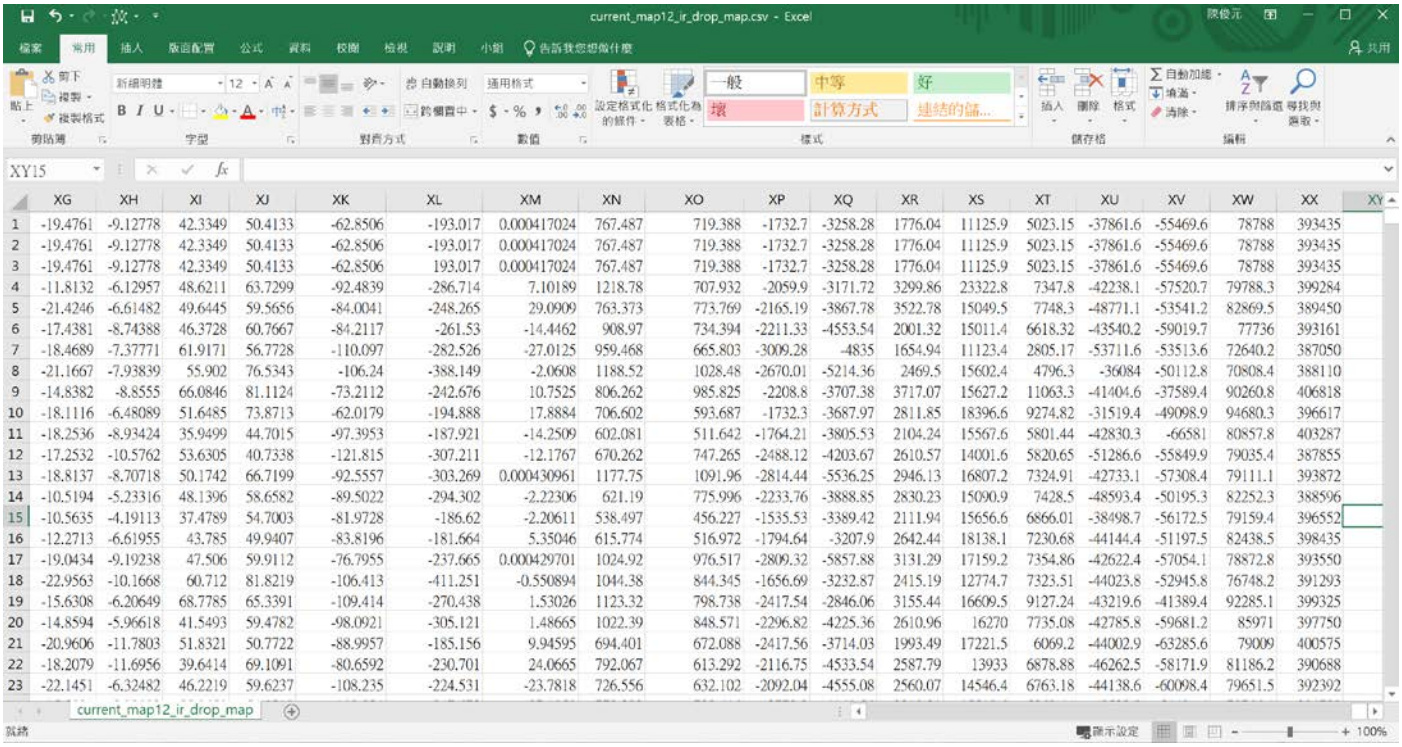
fake-circuit-data/current_map96_ir_drop_map.csv

Below is a screenshot snippet of map12_ir_drop_map.csv (label).

This appears to be an unusually high IR drop value.

I am curious to know if this is a bug, or if we are indeed expected to handle such cases.

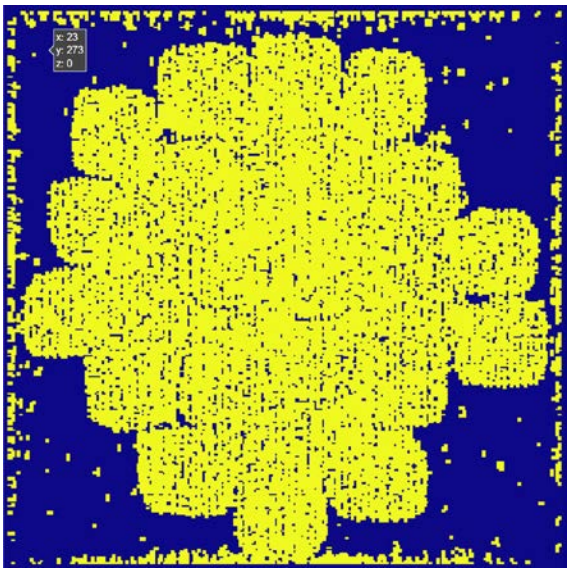
Thank you for your attention to this matter.



	XG	XH	XI	XJ	XK	XL	XM	XN	XO	XP	XQ	XR	XS	XT	XU	XV	XW	XX
1	-19.4761	-9.12778	42.3349	50.4133	-62.8506	-193.017	0.000417024	767.487	719.388	-1732.7	-3258.28	1776.04	11125.9	5023.15	-37861.6	-55469.6	78788	393435
2	-19.4761	-9.12778	42.3349	50.4133	-62.8506	-193.017	0.000417024	767.487	719.388	-1732.7	-3258.28	1776.04	11125.9	5023.15	-37861.6	-55469.6	78788	393435
3	-19.4761	-9.12778	42.3349	50.4133	-62.8506	193.017	0.000417024	767.487	719.388	-1732.7	-3258.28	1776.04	11125.9	5023.15	-37861.6	-55469.6	78788	393435
4	-11.8132	-6.12957	48.6211	63.7299	-92.4839	-286.714	7.10189	1218.78	707.932	-2059.9	-3171.72	3299.86	23322.8	7347.8	-42238.1	-57520.7	79788.3	399284
5	-21.4246	-6.61482	49.6445	59.5656	-84.0041	-248.265	29.0909	763.373	773.769	-2165.19	-3867.78	3522.78	15049.5	7748.3	-48771.1	-53541.2	82869.5	389450
6	-17.4381	-8.74388	46.3728	60.7667	-84.2117	-261.53	-14.4462	908.97	734.394	-2211.33	-4553.54	2001.32	15011.4	6618.32	-43540.2	-59019.7	77736	393161
7	-18.4689	-7.37771	61.9171	56.7728	-110.097	-282.526	-27.0125	959.468	665.803	-3009.28	-4835	1654.94	11123.4	2805.17	-53711.6	-53513.6	72640.2	387050
8	-21.1667	-7.93839	55.902	76.5343	-106.24	-388.149	-2.0608	1188.52	1028.48	-2670.01	-5214.36	2469.5	15602.4	4796.3	-36084	-50112.8	70808.4	388110
9	-14.8382	-8.8555	66.0846	81.1124	-73.2112	-242.676	10.7525	806.262	985.825	-2208.8	-3707.38	3717.07	15627.2	11063.3	-41404.6	-37589.4	90260.8	406818
10	-18.1116	-6.48089	51.6485	73.8713	-62.0179	-194.888	17.8884	706.602	593.687	-1732.3	-3687.97	2811.85	18396.6	9274.82	-31519.4	-49098.9	94680.3	396617
11	-18.2536	-8.93424	35.9499	44.7015	-97.3953	-187.921	-14.2509	602.081	511.642	-1764.21	-3805.53	2104.24	15567.6	5801.44	-42830.3	-66581	80857.8	403287
12	-17.2532	-10.5762	53.6305	40.7338	-121.815	-307.211	-12.1767	670.262	747.265	-2488.12	-4203.67	2610.57	14001.6	5820.65	-51286.6	-55849.9	79035.4	387855
13	-18.8137	-8.70718	50.1742	66.7199	-92.5557	-303.269	0.000430961	1177.75	1091.96	-2814.44	-5536.25	2946.13	16807.2	7324.91	-42733.1	-57308.4	79111.1	393872
14	-10.5194	-5.23316	48.1396	58.6582	-89.5022	-294.302	-2.22306	621.19	775.996	-2233.76	-3888.85	2830.23	15090.9	7428.5	-48593.4	-50195.3	82252.3	388596
15	-10.5635	-4.19113	37.4789	54.7003	-81.9728	-186.62	-2.20611	538.497	456.227	-1535.53	-3389.42	2111.94	15656.6	6866.01	-38498.7	-56172.5	79159.4	396552
16	-12.2713	-6.61955	43.785	49.9407	-83.8196	-181.664	5.35046	615.774	516.972	-1794.64	-3207.9	2642.44	18138.1	7230.68	-44144.4	-51197.5	82438.5	398435
17	-19.0434	-9.19238	47.506	59.9112	-76.7955	-237.665	0.000429701	1024.92	976.517	-2809.32	-5857.88	3131.29	17159.2	7354.86	-42622.4	-57054.1	78872.8	393550
18	-22.9563	-10.1668	60.712	81.8219	-106.413	-411.251	-0.550894	1044.38	844.345	-1656.69	-3232.87	2415.19	12774.7	7323.51	-44023.8	-52945.8	76748.2	391293
19	-15.6308	-6.20649	68.7785	65.3391	-109.414	-270.438	1.53026	1123.32	798.738	-2417.54	-2846.06	3155.44	16609.5	9127.24	-43219.6	-41389.4	92285.1	399325
20	-14.8594	-5.96618	41.5493	59.4782	-98.0921	-305.121	1.48665	1022.39	848.571	-2296.82	-4225.36	2610.96	16270	7735.08	-42785.8	-59681.2	85971	397750
21	-20.9606	-11.7803	51.8321	50.7722	-88.9957	-185.156	9.94595	694.401	672.088	-2417.56	-3714.03	1993.49	17221.5	6069.2	-44002.9	-63285.6	79009	400575
22	-18.2079	-11.6956	39.6414	69.1091	-80.6592	-230.701	24.0665	792.067	613.292	-2116.75	-4533.54	2587.79	13933	6878.88	-46262.5	-58171.9	81186.2	390688
23	-22.1451	-6.32482	46.2219	59.6237	-108.235	-224.531	-23.7818	726.556	632.102	-2092.04	-4555.08	2560.07	14546.4	6763.18	-44138.6	-60098.4	79651.5	392392

A18. Thanks this is fixed with the 6/13 release of the updated benchmarks. Please let me know if you have any more questions.

Q19. I would like to ask about problem C.
In the problem description file, it's said that the goal of IR drop simulation is to find the voltage at the nodes of the PDN that connect to the instances (current sources, the yellow part in the figure below). But we found that there are some nodes in the bottom layer of the chip that are not connected to the current sources(the blue part in the figure below).
Should we predict the IR drop on these nodes without current sources on the bottom layer(seems useless)? Or do we just need to give the values where there is a current source in the output .csv file? If it is the former, it will bring us difficulties in implementation.



A19. Thanks for the question. Yes, there will be some nodes that do not have current sources because there may not be instances nearby. Using commercial tools (golden solvers), the nodes will have some voltage value even if you do not have current sources. The output.csv should have a value of IR drop for every pixel in the matrix. Ultimately, we will be comparing a pixel wise difference as stated in evaluation section. So if you predict the voltage at every node, but then

Q20. Could you please help check the following questions? Thank you.
1) What will be the size of the largest chip in the hidden test case?
2) When will more fake data be provided?
A20. Here are the answers for these questions.
1) The hidden testcases can be of any size. The model should be chip area independent for transferability. This is one of the challenges of the problem. However, I can say that in the hidden testcases the largest chip size is not significantly different from the largest chip size in the training (fake and real circuit data) data.
2) After tremendous consideration, we decided that no more fake data will be provided. It is difficult to look through 1000's of files to check correctness with the power grid. So we are sticking to hundred datapoints for simplicity in dataset generation and bug fixing.

Q21. We have some questions about problem C. The information of the current source given by the netlist includes its coordinates. We extract it to form a matrix. We want to know whether the current map is generated by this matrix or is generated separately using a PDN analysis tool?
Looking forward to your early reply.
A21. The current map is generated using a PDN analysis tool. For the fake data it is generated using GANs and real circuit data it is generated from OpenROAD. The spice file is then generated from the current maps.

Q22. According to the previous QA mentioned, we were expecting to receive approximately 1000 additional fake benchmarks by June 16th. Could you please help advise if any progress of this data and when we can expect to receive it?
Thank you for your attention to this matter.
A22. After tremendous consideration, we decided that no more fake data will be provided. It is difficult to look through 1000's of files to check correctness with the power grid. So, we are sticking to hundred datapoints for simplicity in dataset generation and bug fixing.

Q23. I am writing to inquire about the IR drop map provided in the fake and real data.
In the problem statement page. 1um is considered as the width and height of a pixel. For the coordinate in spice file, 2000 = 1um.
According to our observation, some cases have mismatched pixel maps between the spice file and the map.csv file.
For example, in the real testcase3, the x and y coordinate range is both (0,1857600), which converted to um is (0um, 928.8um). To get the range we choose the smallest range that could cover all nodes' location in the spice file
To our understanding, the pixel map should be 929 * 929. But the offered current_map/pdn_density/eff_dist_map/ir_drop_map.csv are all 930 * 930.

So we wonder if the coordinate conversion method is different, or if the chip core area is not the range covering all nodes in the spice file.

For your reference, we have summarized the mismatched case list as shown below:
(case name) (offered size) (calculated size)
real----------03 (930, 930) (929, 929)
real----------04 (930, 930) (929, 929)
real----------17 (566, 566) (565, 565)
real----------18 (566, 566) (565, 565)
fake----------01 (547, 547) (545, 545)
fake----------05 (690, 690) (689, 689)
fake----------06 (527, 527) (526, 526)
fake----------07 (890, 890) (889, 889)
fake----------08 (726, 726) (725, 725)

fake----------09 (510, 510) (509, 509)
fake----------10 (767, 767) (766, 766)
fake----------15 (880, 880) (879, 879)
fake----------16 (595, 595) (593, 593)
fake----------17 (741, 741) (740, 740)
fake----------19 (631, 631) (630, 630)
fake----------20 (851, 851) (850, 850)
fake----------21 (422, 422) (421, 421)
fake----------22 (763, 763) (761, 763)
fake----------23 (230, 230) (229, 229)
fake----------25 (570, 570) (569, 569)
fake----------29 (882, 882) (881, 881)
fake----------30 (280, 280) (279, 279)
fake----------31 (602, 602) (601, 601)
fake----------33 (842, 842) (841, 841)
fake----------35 (278, 278) (277, 277)
fake----------36 (482, 482) (481, 481)
fake----------37 (319, 319) (317, 317)
fake----------40 (611, 611) (610, 610)
fake----------42 (268, 268) (267, 267)
fake----------43 (570, 570) (569, 569)
fake----------44 (688, 688) (687, 687)
fake----------45 (875, 875) (874, 874)
fake----------46 (568, 568) (567, 567)
fake----------49 (234, 234) (233, 233)
fake----------52 (904, 904) (903, 903)
fake----------54 (429, 429) (428, 428)
fake----------56 (470, 470) (469, 469)
fake----------57 (251, 251) (250, 250)
fake----------60 (330, 330) (329, 329)
fake----------61 (511, 511) (509, 509)
fake----------63 (534, 534) (533, 533)
fake----------65 (330, 330) (329, 329)
fake----------67 (772, 772) (771, 771)
fake----------69 (242, 242) (241, 241)
fake----------71 (895, 895) (893, 893)
fake----------72 (895, 895) (893, 893)
fake----------74 (347, 347) (346, 346)
fake----------76 (614, 614) (613, 613)
fake----------77 (247, 247) (245, 245)
fake----------78 (736, 736) (735, 735)
fake----------82 (724, 724) (723, 723)

fake----------84 (679, 679) (677, 677)
fake----------87 (650, 650) (649, 649)
fake----------89 (827, 827) (826, 826)
fake----------90 (679, 679) (677, 677)
fake----------92 (530, 530) (529, 529)
fake----------93 (897, 897) (896, 896)
fake----------95 (530, 530) (529, 529)
fake----------97 (304, 304) (303, 303)

Thank you for your attention to this inquiry.

A23. Thank you for pointing this mismatch. This is basically a quantization issue based on my understanding. There will never be a situation where the mismatch is more than 2. The way the benchmark generation code is created makes it hard to overcome this challenge. Participants can use the size of the current map as the size of the chip. If you are using graph techniques to predict IR drop then you can use the IR drop of the nearest neighboring node for the additional last two columns and rows of the csv as a suggestion. We will consider this in the evaluation. So for example, if the calculated column is 677x677 however, the output of the original map (current map csv) is 679x679 use the IR drop value of the nearest row and column for those extra two rows and two columns, i.e., just copy the values over. We will be sure to modify our evaluation scripts accordingly. Hope this helps!

Q24. I have some questions regarding Problem C. Firstly, I am unsure whether the checkpoint file we submit must be named as 'chk_pnt.file'. I am not sure what the '.file' suffix means, as the checkpoint file in the PyTorch framework is usually named with the '.pt' extension. Should we perform a transformation from '.file' to '.pt' or some other format?

Secondly, will the time taken to read the 'chk_pnt.file' be included in the overall inference time? I am concerned that if it is included, the time taken could exceed 2 seconds even if our solution is a machine learning model.

Thank you for your time.

A24. The format of your file does not matter. .pt is okay. As long as your binary or source code can read your checkpoint that is all that matters. We have removed the 2s runtime constraint.

Q25. I have some problems about the alpha test of problem C.

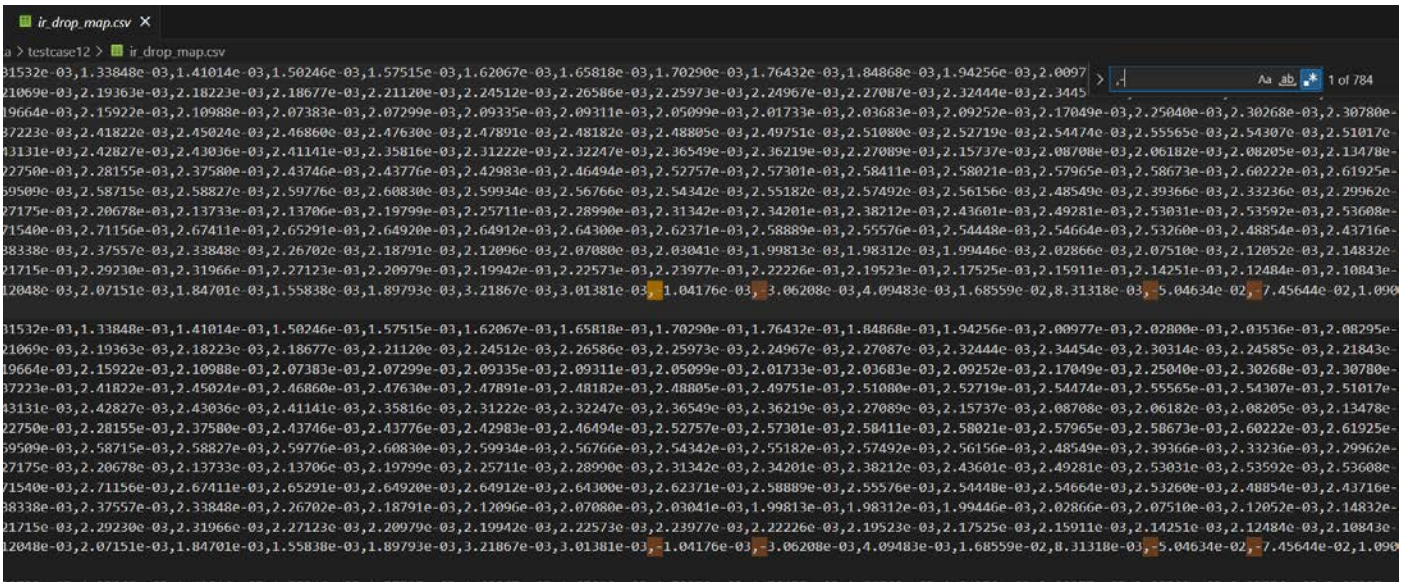
1. We may want more instructions on how to pack our script. We use pyinstaller to pack our script, but the resulting executable file is over 3 GB, and runs more than 10 times slower than the source code, because the dependent libraries are also packed into the executable file. If we got it right, pyinstaller (similar for py2exe and cx_freeze) will inevitably packed the dependent libraries, so that the resulting executable file can run independent of local environment. But this is unfavourable for the evaluation due to the prolonged runtime, so are there any other ways to pack our scripts?

Thank you very much for your time and attention.

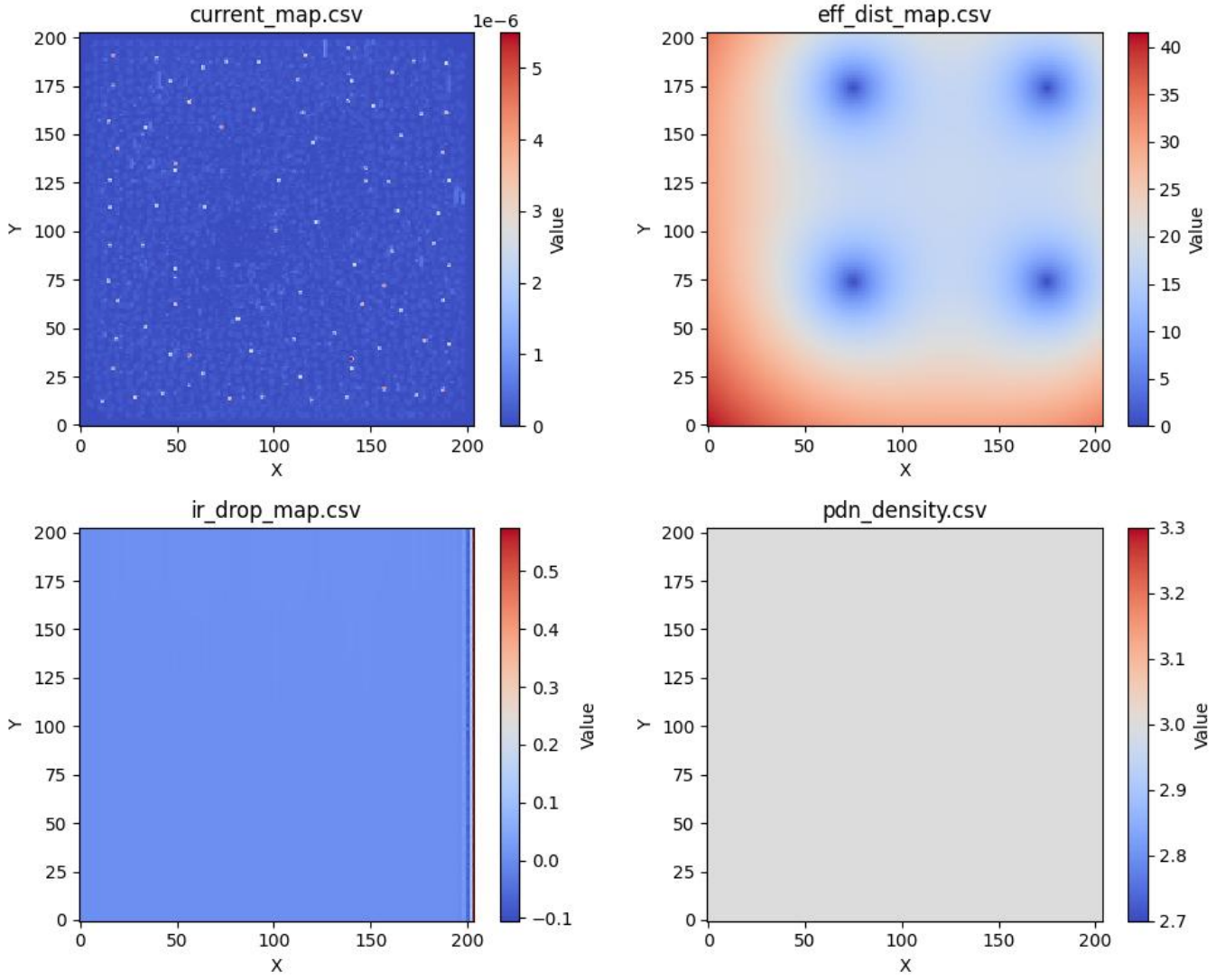
A25. The format of your file does not matter. .pt is okay. If your binary or source code can read your checkpoint that is all that matters. We have removed the 2s runtime constraint.

Q26. I would like to ask a question of ICCAD ProblemC. Thank you.
In the problem description, the time limit is set to 2 seconds. However, 2 seconds of data processing is not enough. Should we divide the data processing and model inference in two steps (different code)?
A26. We have removed the 2s runtime constraint. You can create only one executable so that we have apples to apples comparisons across all participants.

Q27. I am writing to inquire about the IR drop map provided in the fake and real data.
In my understanding, The value in the IR drop map refers to the voltage reduction in the position compared to the voltage source. So a negative value means that the voltage here is even higher than the voltage source, which should not happen in netlist with only resistance, voltage sources, and current source.
According to our observation, the testcase12 in the real data has some negative value in the IR drop map, so as do some other cases in the fake data.



For better observation, we plot the four maps in testcase12 as shown below:



The negative values all appear at the right edge.

So we wonder about the meaning of this negative data. Is our understanding of the IR drop map wrong or is there any other factor that could lead to the voltage being higher than the voltage source?

Thank you for your attention to this inquiry. We look forward to your kind response and appreciate your support.

A27. These issues were fixed with the 6/12 and 6/22 updates.

Q28. I have a question for Organizers of Problem C: Static IR Drop Estimation Using Machine-Learning.

> (b) F1 score: A binary classification metric that uses 10% of the maximum IR drop of each benchmark as the threshold to perform classification.
>
> $F1\ score = 2 * Precision * Recall / (Precision + Recall)$
> $Precision = TP / (TP + FP)$
> $Recall = TP / (TP + FN)$
>
> Where TP = True positive, FP = False positive, TN = True negative, and FN = False negative.
> The positive class is nodes with the top 10% of IR drop. The goal is to have a high F1 score.

In the evaluation of F1 score, I'm unable to understand how TP, FP, TN and FN are categorized. I'm not sure which values constitutes to a False. For example, assume a 'testcase-A', if the maximum IR drop of

testcase-A is 0.50; And predicted maximum IR drop of the generated map is within 10% of maximum testcase-A IR Drop (0.45,0.55), for that testcase it is True. If predicted maximum value is not in range, for that testcase it is False. Could you please elaborate how TP, FP, TN and FN are categorized.
Thanks for your help.
A28. With the 6/22 update this metric has been made clear. Please take a look at Section 4 of the PDF and let me know if you have more questions. The section also has an example.

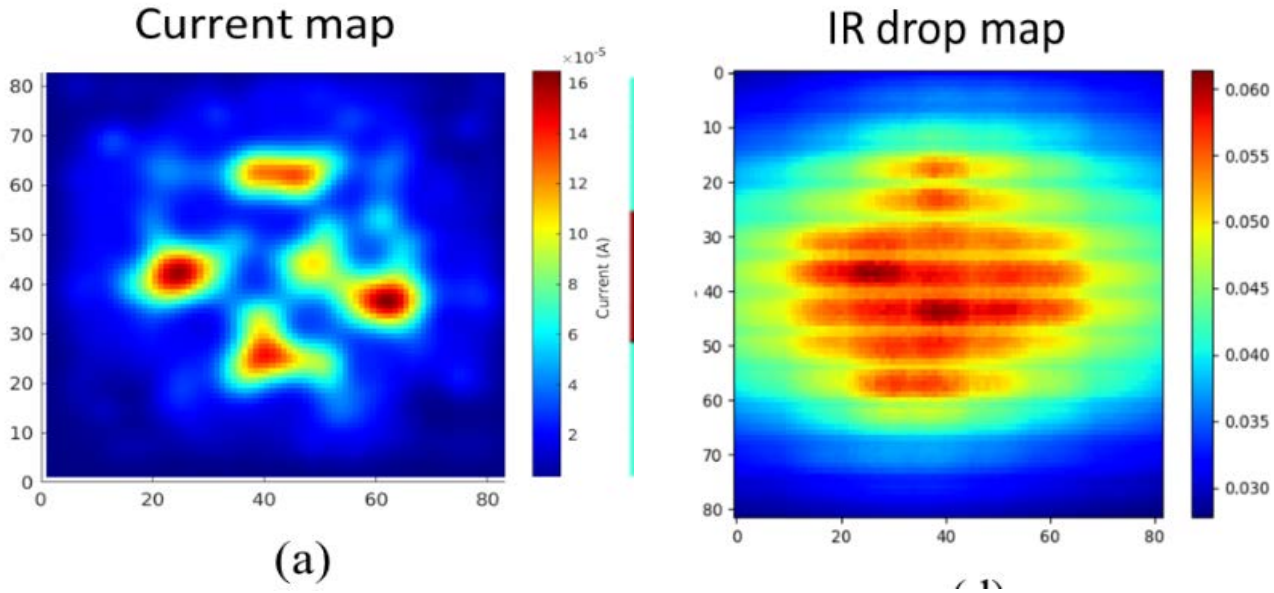
Q29. We have some questions about problem C.
Could you please provide us with information about the chip process node corresponding to the competition data set? Is it base on 12nm or another size? It's important to us.
Thank you for your attention to this matter.
A29. All the designs are based on NANGATE45.

Q30. We have some questions about problem C. We want to know how the .sp file is generated by the current map. Can you provide us with more information about it? It means a lot to us. Thank you.
A30. The translation from sp to csv was covered in an earlier answer, reproduced below for your convenience:

"The locations in the spice file can be converted to the csv file by using a 2000dbu. For instance, 113400 is the .sp file location 113400/2000 = 56.7 um. The matrix in the .csv files is at a 1um granularity. So in the above example, the 821x820 has come by dividing the node in the resistance you highlighted by 2000. I have updated the contest document to reflect this.""

Note that the sp file cannot be generated directly from the current map as it also needs information about the grid. The csv file can be created from the sp file as described above.

Q31. We would like to knowknow the specific reference relationship of the pixels in the csv file.
For example, if a chip size is 80x80, the first value of the csv file refers to the [0,0]-[1,1] area of the chip, or the [0,80], [1,79] area.
In other words, is the starting coordinate (0,0) of the x-axis and y-axis at the bottom-left or top-left corner of the image? (The reason for this question is that we noticed the coordinate origin of the current map and ir drop map given in the problem statement are not in the same place)."



Thank you.

A31. The pixels of the CSV follow the same order in both the IR drop and current maps. The 0,0 index in the csv (when reading the csv through numpy) refer to the 0,0 location within the for both current maps and IR drop maps. The image in the document is just an example and does not directly reflect the csv.

Q32. I am writing to request your advice on a technical issue that I am currently facing. I am attempting to install Pytorch Geometric on my server, however, I have encountered a problem related to an insufficient gcc version.

I was wondering if you could recommend a specific version of Pytorch Geometric that is compatible with the server setup.

Your assistance would be highly appreciated.

A32. Note form CAD Contest: The current version is gcc 4.8.5 for your reference. And it would not be changed due to the environment need to be remained the same of the beta test submission and the final submission.

Q33. I am writing to inquire about the evaluation process.

1) Evaluation Criteria: We have been informed that the 2 seconds constraint for Problem C has been removed,. However, we would like to confirm whether runtime is still a factor in the evaluation process, as the specifications mentioned that runtime would account for 10% of the final score. We noticed that during the alpha testA, the runtime of each test case was not provided. Could you kindly explain the reason behind this discrepancy or clarify whether runtime is still considered in the final evaluation?

2) Submission Format:

In the alpha test results provided, there was a comment stating, "**Optionally, if submitting the executable harder, if you are okay with providing the scripts, please share the python script directly.**" We want to ensure that we understand the submission requirements correctly. Is it acceptable to submit the .py files directly instead of the binary executable files, given that there may be a substantial difference in runtime

between the two formats? If this is feasible, we will also include a detailed README file along with the necessary scripts to run the .py files smoothly for the coming final submission.
Thank you for taking the time to read and address our inquiries.
A33. Please have the following reply:
1) The constraint on runtime is removed but it is still worth 10% of the total evaluation. The runtimes will be evaluated in the beta submission.
2) For calculation of runtimes, we will only consider the time taken by the executable. This is to ensure a fair comparison between different teams. However, we did see multiple teams have issues with the executable. As such we provide an option to give a python script that can be evaluated for the F1 score and MAE at the very least. Any runs from the python script will only count towards the MAE and F1 scores but will get a 0 in their score for runtime if the executable does not run.

Q34. I have a couple of questions pertaining to Problem C of the competition.
1) I would like to clarify whether the evaluation process for Problem C will utilize both the GPU and CPU, or if it will exclusively rely on the CPU.
2) Regarding the submission of the Requriements.txt file, which is expected to be installed using "pip3 install requirements.txt" in the organizers' virtual environment, I was wondering if the GitHub-released package which require a different installation command can be installed.
Thank you for your time and assistance in addressing these queries. Your guidance is greatly appreciated.
A34. Please have the following reply:
1) The main evaluation will be performed on the TSRI machines that are CPU only to make it equally accessible to all. The participants code should not be hardcoded to use a GPU.
2) We will only support general python libraries that can be installed via pip. Installing custom GitHub based programs would be difficult to support at evaluation.

Q35. I am writing to inquire about Problem C.
1) Could you please tell me the results of our beta test submission? We need the results to refine our algorithm. Our team ID is cadc1023. Thanks.
2) Does the test submission employ CUDA to compute the results?
3) Does the maximal value for computing F1 from the actual IR drop matrix or the predicted IR drop matrix?

For example, predicted IR drop matrix is:

4.43E-03 5.83E-03

6.00E-03 1.04E-03

Actual IR drop matrix is:

4.63E-03 5.23E-03

5.93E-03 0.04E-03

When computing F1, the maximal value is 6.00E-03 or 5.93E-03?
Thanks for your time and help.
A35. Please have the following reply:

1) The results for the Beta submission have been released on the website.
2) The main evaluation will be performed on the TSRI machines that are CPU only to make it equally accessible to all. The participants code should not be hardcoded to use a GPU.
3) The maximal value is always taken from the actual IR drop matrix and is used as the reference to define the classes.