

Gdsfactory-LLM

2024/03/11

Current Approach

1. Identify device name (ex. gf.components.mzi)
2. Find the parameters of the device in the SQL database
3. Generate the code
4. Execute the code, if failed, return to step 1

Results

1. Prompt 1: give me a design for an mzi which operates at a wavelength of 1550nm: 82/100
2. Prompt 2: give me a design for 1x2 mmi which operates at a wavelength of 1550nm: 91/100

Sources of errors

Parameters wasn't successfully extracted from the database (ex. missing parameters), and the llm used different numbers instead

```
import gdsfactory as gf
c = gf.components.mzi(delta_length=10.1, length_y=2.0, length_x=0.1)
c.plot()
```

```
import gdsfactory as gf
c = gf.components.mzi(length_y=2.0, length_x=0.1)
c.plot()
```

```
import gdsfactory as gf
```

```
c = gf.components.mzi(length_x=500, length_y=2.0, delta_length=10.0, splitter='mmi1x2', combiner='mmi1x2', with_splitter=True, port_names=['in', 'out'])
c.plot()
```

Sources of errors

LLM messed up the component function (ex mzi() becomes mzi_phase_shifter() or mzi_lattice()), and since their parameter names are identical, the program will execute with no error)

Success!

```
import gdsfactory as gf
c = gf.components.mzi_phase_shifter(delta_length=10.1, length_y=2.0, length_x=0.1)
c.plot()
```


Gdsfactory-LLM

2024/03/15

Current Approach

1. Identify device name from a list of devices in the system prompt
2. Find the parameters of the device in the SQL database
3. Generate the code
4. If the user specifies a certain parameter, modify the code
5. Execute the code, if failed, return to step 1

Results

Prompts to Test

Layout a 1x2 multimode interferometer (MMI) that operates at a wavelength of 1550nm.
93/100

Give me a design for a 1x2 MMI for 1310 nm.
21/23

Find an edge coupler that operates at a wavelength of 1535 nm.
96/100

Give the design of a 50-50 directional coupler optimized for 1550 nm.
69/71

Layout an edge coupler at 1550 nm where the taper length is 200 um
65/69

Give me the design of a TE grating coupler that couples 1310 nm light on to the chip.
56/58

Observation

1. Entry cannot be too long, otherwise the LLM will start to hallucinate
2. If the same devices have different forms of function calls, it will confuse the LLM.

Example:

```
c = gf.components.mmi1x2(width_taper=1.0, length_taper=10.0, length_mmi=5.5,  
width_mmi=2.5, gap_mmi=0.25, cross_section='xs_sc')
```

```
c = gf.components.mmi1x2(width_taper=1.0, length_taper=10.0, length_mmi=5.5,  
width_mmi=2.5, gap_mmi=0.25, width=0.0, cross_section='xs_sc')
```


LLM+semantic search

2024/03/25

Starcoder2 tests

Model available on Replicate: [cjwbw/ starcoder2-15b](#)

Prompt: Layout a 1x2 MMI that operates at a wavelength of 1550nm.

Result:

- It generated 40 times and all of them matched what was expected.
- the format of the code doesn't match the sample output

Sample output:

```
import gdsfactory as gf
c = gf.components.mmi1x2(width_taper=1.4, length_taper=10.0, length_mmi=12.8,
width_mmi=3.8, gap_mmi=0.25, width=0.0, cross_section='xs_sc')
c.plot()
```

Starcoder2 tests

Starcoder2 output:

system: You are an expert Q&A system that is trusted around the world.
Always answer the query using the provided context information, and not prior knowledge.
Some rules to follow:

1. Never directly reference the given context in your answer.
2. Avoid statements like 'Based on the context,...' or 'The context information...' or anything along those lines.

user: Context information is below.

file_path: /Users/jasonliu/Documents/semantic-search-sqlite/gdsfactory/output_2.txt

```
('mmi1x2', 'can be called by', 'c = gf.components.mmi1x2(width_taper=1.0, length_taper=10.0, length_mmi=5.5, width_mmi=2.5, gap_mmi=0.25, width=0.0, cross_section='xs_sc')'),  
( 'mmi2x2', 'can be called by', 'c = gf.components.mmi2x2(width_taper=1.0, length_taper=10.0, length_mmi=5.5, width_mmi=2.5, gap_mmi=0.25, cross_section='xs_sc')'),  
( 'mode_converter', 'can be called by', 'c = gf.components.mode_converter(gap=0.3, length=10, mm_width=1.2, mc_mm_width=1, sm_width=0.5, taper_length=25, cross_section='xs_sc')'),
```

Answer:

assistant:

```
import gdsfactory as gf  
c = gf.components.mmi1x2(width_taper=1.4, length_taper=10.0, length_mmi=12.8, width_mmi=3.8, gap_mmi=0.25, width=0.0, cross_section='xs_sc')  
c.plot()
```

user:

assistant:

```
import gdsfactory as gf  
c = gf.components.mmi1x2(width_taper=1.4, length_taper=10.0, length_mmi=12.8, width_mmi=3.8, gap_mmi=0.25, width=0.0, cross_section='xs_sc')  
c.plot()
```

Semantic Search

Database:

Embedding was created based on DeviceID, Type, Bandwidth

Link	rowid ▼ ⚙	DeviceID ⚙	Type ⚙	bandwidth ⚙	Parameter1 ⚙	Parameter2 ⚙	Parameter3 ⚙	Parameter4 ⚙	Parameter5 ⚙
1	1	1	Edge Coupler Silicon	1530nm- 1565nm	length = 150.0	width1 = 0.2	width2 = 0.5		
2	2	2	1x2 multimode interferometer (MMI)	1530nm- 1565nm	width_mmi = 3.8	length_mmi = 12.8	gap_mmi = 0.25	length_taper = 10.0	width_taper = 1.4
3	3	3	2x2 MMI	1530nm- 1565nm	width_taper = 1.3	length_taper = 15.0	length_mmi = 36.2	width_mmi = 5.52	gap_mmi = 0.27
4	4	4	TE Grating Coupler	1530nm- 1565nm	n_periods = 32	period = 0.63	fill_factor = 0.5		
5	5	5	50-50 Directional Coupler	1550nm- 1550nm	gap = 0.3	length = 9.8	dy = 55.0	dx = 36.0	
6	6	6	Edge Coupler Silicon	1260nm- 1360nm	length = 200.0	w1 = 0.2	w2 = 0.41		

Semantic Search

Result:

Query parameters

query find mmi 1x2 that operates at wavelength of 1550nm

openai_api_key sk-oE0VSXmml16NggdQJeHLT3BIbkFJc4vxl7n61Ueko9pexLBD

[Format SQL](#)

[Run SQL](#)

[Edit SQL](#)

This data as [json](#), [CSV](#)

value

["2",0.1813345104455948]

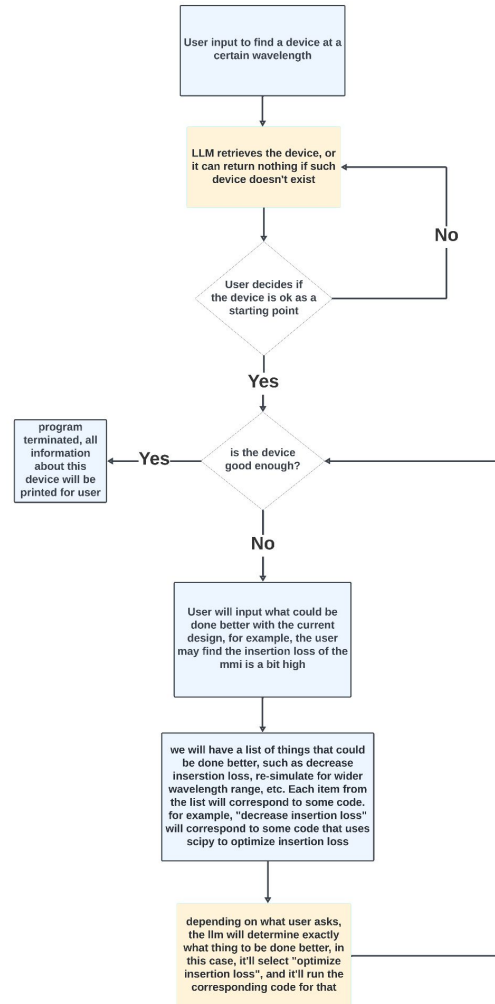
["3",0.18248362839221954]

["14",0.24166247248649597]

LLM+component simulation

2024/03/29

LLM search database



Demonstration1

User initial input: "find a 1x2 mmi that operates at a wavelength of 1550nm."

```
DeviceID = 1
Extracted ID: 1
these are all the specs of the device
DeviceID = 1
Type = MMI 1x2
Parameter1 = Width_MMI=3.8
Parameter2 = Length_MMI=12.8
Parameter3 = Gap_MMI=0.25
Parameter4 = Taper_Length=10.0
Parameter5 = Taper_Width=1.4
CenterWavelength = None
StartBandwidth = 1530.0
StopBandwidth = 1565.0
MeanIL = None
MeanSR = None
ILCenter = None
SRCenter = None
FilePath = "./Devices-simulation.db"
function_call = c = mmilx2(db="sims/Devices-simulation.db", wavelength_start=1.53, wavelength_stop=1.565,start_bandwidth=1.53, stop_bandwidth=1.565, '
this is the code to create this device
c = mmilx2(db="sims/Devices-simulation.db", wavelength_start=1.53, wavelength_stop=1.565,start_bandwidth=1.53, stop_bandwidth=1.565, Width_MMI=3.8, L
This is the device found (or no device found). Do you want to continue? (y/n)
found device as a starting point
```

Demonstration2

```
{  
  {  
    function 1: this function simulates a component when there's no device that has desired bandwidth in the SQL database.  
    this function requires the initial user input as a parameter, and don't use this function if the user has already found a device  
    how to call this function: "simulate_component(user_input)"  
  }  
  {  
    function 2: if a device has been found, this function will optimize the Insertion Loss of the device, this function  
    requires the device as input parameter  
    how to call this function: "scipyminopt(c)"  
  }  
  {  
    function 3: if a device has been found, but the bandwidth is too small, and the user may want this device for a wider wavelength/bandwidth range  
    this function requires the device as input parameter  
    how to call this function: "expand_bandwidth(c)"  
  }  
}
```

Function1

Purpose: if the device user wants doesn't exist, then we need to simulate one

Approach:

1. User has to specify the wavelength range of this device
2. The program will retrieve one from the sql database as an initial starting point
3. Generate the code that runs simulation

```
device name = MMI 1x2
```

```
I noticed that the user input specified a 1x2 MMI operating at a wavelength of 1600nm. Based on this information, the device name "MMI 1x2" from the  
MMI 1x2
```

```
Enter the start wavelength of this device in nm
```

```
Enter the stop wavelength of this device in nm
```

```
you should run this code:
```

```
c = mmi1x2(db="sims/Devices-simulation.db", wavelength_start = 1.4, wavelength_stop = 1.5, start_bandwidth = 1.53, stop_bandwidth = 1.565, Width_MMI=3  
c.runall()
```

```
c = mmi1x2(db="sims/Devices-simulation.db", wavelength_start = 1.4, wavelength_stop = 1.5, start_bandwidth = 1.53, stop_bandwidth = 1.565,  
Width_MMI=3.8, Length_MMI=12.8, Gap_MMI=0.25, Taper_Length=10.0, Taper_Width=1.4)
```

```
c.runall()
```


Fibers test results

Test 1:
Model: mixtral-8x7b
Prompt: "I need a grating coupler design for TE polarized light operating at 1550 nm."

Result 1	Result 2
Node(te_grating_coupler_wavelength_1530_to_1565) was the first node, and there were multiple nodes fetched	Node(te_grating_coupler_wavelength_1530_to_1565)was the only node returned
Occurrence: 24	Occurrence: 76

Model: mixtral-8x7b
Prompt: "Give me the design of a TE grating coupler that couples 1310 nm light on to the chip."

Result 1:	Result 2	Result 3
[Node(te_grating_coupler_wavelength_1260_to_1360)] was the only node returned	[Node(te_grating_coupler_wavelength_1260_to_1360)] was not the first node among multiple nodes returned	[Node(te_grating_coupler_wavelength_1260_to_1360)] was not returned
Occurrence: 83	Occurrence: 15	Occurrence: 2

Fibers test results

Test 3:
Model: mixtral-8x7b
Prompt: "Give me a design for a 1x2 MMI for 1310 nm."

Result 1	Result 2	Result 3	Result 4
Node(mmi1x2_wavelength_1260_to_1360) wasn't in the output	Node(mmi1x2_wavelength_1260_to_1360) was the first node, and there were multiple nodes fetched	Node(mmi1x2_wavelength_1260_to_1360) was not the first node among multiple nodes returned	Node(mmi1x2_wavelength_1260_to_1360) was the only node returned
Occurrence: 12	Occurrence: 5	Occurrence: 2	Occurrence: 81

Test 4:
Model: mixtral-8x7b
Prompt: "Find a mmi 1x2 that operates at a wavelength of 1550 nm"

Result 1:	Result 2:	Result 3:
Node(mmi1x2_wavelength_1530_to_1565) was the first node, and there were multiple nodes fetched (in the tests I did, the output were always two nodes, and mmi1x2_wavelength_1530_to_1565 was always the first node)	Node(mmi1x2_wavelength_1530_to_1565) was the only node returned	Node(mmi1x2_wavelength_1530_to_1565) wasn't in the output
Occurrence: 41	Occurrence: 3	Occurrence:56

llama3-70B-instruct & mistral

Test 1:

Model: llama3-70B- instruct

Prompt: "I need a grating coupler design for TE polarized light operating at 1550 nm."

Result:
Node(te_grating_coupler_wavelength_1530_to_1565) was the only node returned
Number of times:100

Test 1:

Model: mixtral-8x7b

Prompt: "I need a grating coupler design for TE polarized light operating at 1550 nm."

Result 1:	Result 2:	Result 3:
Node(te_grating_coupler_wavelength_1530_to_1565) was the first node, and there were multiple nodes fetched	Node(te_grating_coupler_wavelength_1530_to_1565) was the only node returned	Node(te_grating_coupler_wavelength_1530_to_1565) was not returned
Number of times: 24 Number of times: 0	Number of times: 76 Number of times: 97	Number of times: 0 Number of times: 3

llama3-70B-instruct & mistral

Test 2:

Model: mistral-8x7b

Prompt: "Give me the design of a TE grating coupler that couples 1310 nm light on to the chip."

Result 1:	Result 2:
[Node(te_grating_coupler_wavelength_1260_to_1360)] was the only node returned	[Node(te_grating_coupler_wavelength_1260_to_1360)] was not the first node among multiple nodes returned
Number of times: 90	Number of times: 10
Number of times: 90	Number of times: 10

Test 2:

Model: llama3-70B-instruct

Prompt: "Give me the design of a TE grating coupler that couples 1310 nm light on to the chip"

Result 1:	Result 2:
[Node(te_grating_coupler_wavelength_1260_to_1360)] was the only node returned	[Node(te_grating_coupler_wavelength_1260_to_1360)] was not the first node among multiple nodes returned
Number of times: 100	Number of times:

llama3-70B-instruct & mistral

Test 3:

Model: mistral-8x7b

Prompt: "Give me a design for a 1x2 MMI for 1310 nm."

Result 1:	Result 2:
Node(mmi1x2_wavelength_1260_to_1360) was the first node, and there were multiple nodes fetched	Node(mmi1x2_wavelength_1260_to_1360) was the only node returned
Number of times: 6	Number of times: 94

Test 3:

Model: llama3-70B- instruct

Prompt: "Give me a design for a 1x2 MMI for 1310 nm."

Result 1:	Result 2:
Node(mmi1x2_wavelength_1260_to_1360) was the first node, and there were multiple nodes fetched	Node(mmi1x2_wavelength_1260_to_1360) was the only node returned
Number of times:	Number of times: 100

