TCAD Structure Input File Generation Using Large Language Model

Le Minh Long Nguyen
Electrical Engineering Department
San *Jose State University*
San Jose, CA, USA
leminhlong.nguyen@sjsu.edu

Albert Lu
Electrical Engineering Department
San *Jose State University*
San Jose, CA, USA
albert.lu@sjsu.edu

Hiu Yung Wong
Electrical Engineering Department
San *Jose State University*
San Jose, CA, USA
hiuyung.wong@sjsu.edu

***Abstract*— In this paper, we study the possibility of using Large Language Models (LLMs) to create Technology Computer-Aided-Design (TCAD) structure generation input files. LLMs are machine learning models trained on vast amounts of text data from the Web and are designed to understand, generate, and interact with humans through natural languages such as English. However, unlike programming languages which have abundant training examples on the Web, TCAD examples are scarce. In this work, using TCAD Sentaurus Structure Editor (SSE) as an example, 7000 nanowire data are generated to fine-tune open-source models (Llama 2 and 3) to obtain chatbots that can generate an SSE input file for a nanowire with 18 parameters with English as an instruction. Structures are then created using SSE to verify the correctness of the input files. It shows that it is possible to create a chatbot even with limited resources. In-context one-shot learning is also studied. It shows that with large commercial models, in-context one-shot learning is enough to generate the desired SSE input file.**

Keywords—ChatGPT, In-context Learning, Large Language Model (LLM), Llama, One-shot Learning, Technology Computer-Aided Design (TCAD)

I. Introduction

Large Language Models (LLMs) such as ChatGPT have revolutionized various fields from education to medicine. However, they have not been widely used in TCAD. TCAD, as a sophisticated engineering tool, usually requires a significant amount of expertise from the engineers to set up an appropriate structure. It is thus desirable to train LLMs which can create input files for structure generation based on natural languages. However, unlike programming languages such as Python which have abundant training examples on the Web, TCAD examples are scarce.

There are various types of LLMs available, such as Llama [1], ChatGPT [2], and Claude [3]. They all have a transformer architect, but only Llama models are open-source LLMs. Llama has models with parameters from 7 billion to 70 billion. ChatGPT and Claude are estimated to have more than 200 billion parameters.

In this work, using TCAD Sentaurus Structure Editor (SSE) as an example, two aspects of LLMs are studied. Firstly, we studied the possibility of fine-tuning Llama models to generate SSE input files. Secondly, we study the possibility of using in-context one-shot learning on the most powerful LLMs (including ChatGPT and Claude).

II. Data Generation

An SSE nanowire input file is used as a template to generate 7000 pairs of instruction (English description of nanowire) and output (SSE input files). Fig. 1 shows an example. The nanowire is a cascode nanowire [4] with two gate regions under which different dopings and materials are used. The gate oxide thickness, material, and doping in various regions are changed randomly to generate the instruction-output pairs. The data are formatted into the Alpaca dataset format with only instruction and output keys while the "input" key is left empty. Note that the English description in the instruction of the training data is fixed with only the 18 parameters varied. It will be shown that the chatbot to be trained can handle any variation in the English description.

III. LLM Fine-Tuning

A. Fine-Tuning Llama 2 and Llama 3 Models

The data is then used to fine-tune Meta Llama-2 and Llama-3 LLM. Llama-2 LLM models are open-source with 3 sizes of 7

a) Instruction: This structure is a nanowire transistor. There are 5 main regions: the source, short gate, long gate, drain, and oxide. There are also 3 contacts: source, drain, and gate. The gate contact covers the short gate and long gate length. The source region is of the material **SiGe** with a length of **0.32** micrometers and it is doped with **Phosphorus** at a concentration of **633899008937397583872 cm^-3**. The short gate region is of the material **Silicon** with a length of **0.03** micrometers and it is doped with **Arsenic** at a concentration of **6845500531888189603 84 cm^-3.** The long gate region is of the material **Silicon** with a length of **0.07** micrometers and it is doped with **Boron** at a concentration of **708643624252055093248 cm^-3**. The drain region is of the material **Diamond** with a length of **0.32** micrometers and it is doped with **Boron** at a concentration of **666868145841890525184 cm^-3**. The gate oxide thickness is **0.007** micrometers. The nanowire thickness is **0.006** micrometers.

b) Part of the Output:

```
(define tox 0.007)   //   (define Lgs 0.03)  //    (define Lgl 0.07)
(sdegeo:create-rectangle (position XGates Ymin 0)  (position XGatel Ymax 0) "Silicon" "R.Long_Gate")
(sdegeo:create-rectangle (position XGatel Ymin 0)  (position Xmax Ymax 0) "Diamond" "R.Drain")
(sdedr:define-constant-profile      "Const.Source"          Phosphorus          633899008937397583872 )
```

c)

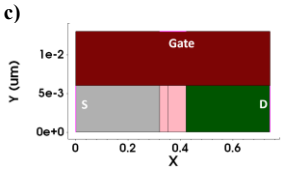


Fig. 1. a) An example of the training data with the 18 parameters highlighted in the natural language instruction. b) A part of the output (SSE input file corresponding to the natural language). c) The corresponding SSE structure (the x and y axes are not in scale).

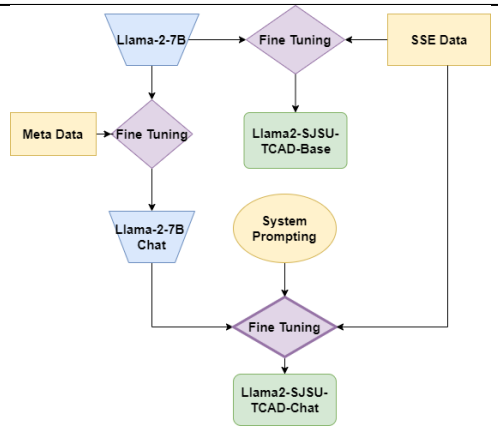


Fig. 2. Evolution of Llama-2-7B models in this study. The models fine-tuned with SSE data are shown in green. A similar flow is used to create Llama3-SJSU-TCAD-Chat.

(Llama-2-7B), 13 (Llama-2-13B), and 70 (Llama-2-70B) billion parameters, while Llama-3 has 2 sizes of 8 (Llama-3-8B) and 70 (Llama-3-70B) billion parameters. Due to the limitation of our computing resources, the Llama-2-7B and Llama-3-8B models are used. Besides the base model for sentence completion/ word prediction, these models are also fine-tuned by Meta to form the corresponding chat versions. In this study, both the base (Llama-2-7B and Llama-3-8B) and chat versions (Llama-2-7B-chat, Llama-3-8B-Instruct) are investigated by fine-tuning with our SSE data (Fig. 2).

The Llama-2-7B base and Llama-2-7B-chat models are fine-tuned using 2 Nvidia GPUs (Quadro RTX 8000) each with 48GB GDDR6 memory. Quantization is used in which 32-bit operations are quantized to 16-bit. The fine-tuning time is 9 and 16 hours for the base and chat model, respectively. In the fine-tuning stage, we followed the training process in "llama-recipes". We need to transform/convert the base model weight/ checkpoint files of Llama-2-7B to Hugging Face weight to use Hugging Face's transformer. We use Low-Rank Adaptation

a) Bad: " <s>[INST] <<SYS>>You are an electrical engineer and writing an input file to Sentaurus Structure Editor to generate the MOSFET structure. Below is the natural language of a semiconductor MOSFET structure description<</SYS>>. Description: \n{description}[/INST]\n\n###respond:{TCAD input file}</s>"

b) Good: "<s>[INST] <<SYS>>You are an intelligent assistant with specialized expertise in electrical engineering and semiconductor device simulation. Your task involves assisting users in creating input files for a TCAD tool, the Sentaurus Structure Editor. This tool plays a critical role in the design and simulation of semiconductor devices, including MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors). When users provide you with a natural language description of a semiconductor MOSFET structure, your job is to translate this description into the structured, precise format required by the Sentaurus Structure Editor. Your translation must accurately reflect the user's specifications, including device type, materials, dimensions, doping concentrations, and any specific configurations or layers they describe. <</SYS>>. Description from user:\n{description}[/INST]\n\n### assistant respond:{TCAD input file}</s>"

Fig. 3. Bad and ineffective (a) vs. good and effective (b) system prompt for Llama-2-7B-chat fine-tuning. The highlighted green texts are believed to help the model achieve more accurate fine-tuning and understand better the meaning of the training data. The highlighted blue texts are special tokens of Llama-2-7B-chat.

<|begin_of_text|><|start_header_id|>system<|end_header_id|>You are an AI with specialized expertise in electrical engineering and semiconductor device simulation. Your primary task is to assist users in creating precise, structured input files for the Sentaurus Structure Editor, a tool essential for the design and simulation of semiconductor devices such as MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors).\nYour role involves the following:\n1. Receive natural language descriptions of MOSFET structures from users.\n2. Accurately translate these descriptions into the structured, precise format required by the Sentaurus Structure Editor.\n3. Ensure the translated file includes all necessary specifications provided by the user, such as device type, materials, dimensions, doping concentrations, and specific configurations or layers.\nThis task requires a deep understanding of semiconductor physics, meticulous attention to detail, and the ability to interpret and formalize technical language.<|eot_id|> <|start_header_id|>user<|end_header_id|> Create a TCAD Sentaurus structure input file based on this instruction: \n{instruction}<|eot_id|>\n\n<|start_header_id|>assistant<|end_header_id|>a ssistant responds: {output} <|eot_id|><|end_of_text|>

Fig. 4. The system prompt in fine-tuning with Llama-3-8B-Instruct special tokens in blue color and data entry in orange. The highlighted green texts are believed to help the model achieve more accurate fine-tuning and understand better the meaning of the training data.

(LoRA) [5] as a Parameter-Efficient Fine-Tuning (PEFT) method to perform the fine-tuning during which the original parameters are fixed and a parallel low-rank adapter network with 4 million parameters is added and trained for this specific application. Note that old and new networks are merged and the original knowledge is preserved in this process. The new base model is dubbed "Llama2-SJSU-TCAD-base". To fine-tune the Llama-2-7B-chat model, we also need to use appropriate system prompting in front of each instruction-output pair. The system prompt should be well-deigned to help the LLM understand the context of the training data during fine-tuning. Fig. 3 shows two system prompts. The second (first) one was effective (ineffective) in training a chatbot to generate correct SSE input files. The new chat model based on the effective prompt is dubbed "Llama2-SJSU-TCAD-chat". 4k data among the 7k generated data are used with 3 epochs. Each epoch has 1000 steps. Since TCAD requires accurate numbers, *temperature* and *top_p*, which control the randomness, are set to be small.

A similar approach is applied to the Llama-3-8B-Instruct model (an updated version of the Llama-2-7B-chat model) which has new special tokens and another well-design system prompt is used (Fig. 4). However, the result was not satisfactory with 4k data and 3 epochs. Therefore, we increased the number of data entries up to 7k with 8 epochs and it took more than 60 hours to finish. Fig. 5 shows the training and validation losses. While the loss is already small after 100 steps of training, good results are only obtained after 8000 steps (i.e. 8 epochs). The best performance finetuned model is named Llama3-SJSU-TCAD-Chat.

B. Performance

A modified English description of the task is shown in Fig. 6a with unseen parameters. It should be noted that uncommonly used English sentence structures and words are deliberately used. New sentences are added and the order of structure

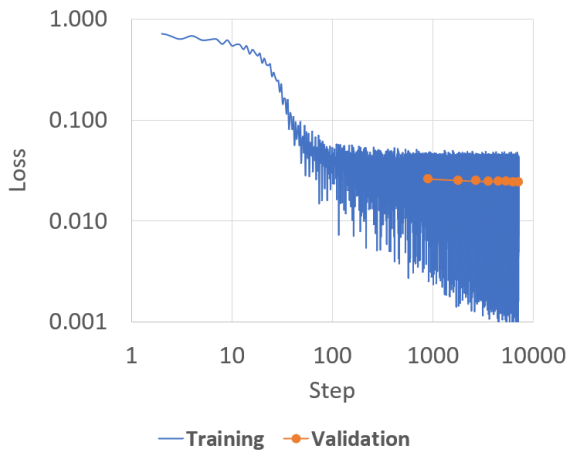


Fig. 5. Training and validation loss as a function of the training step when fine-tuning Llama-3-8B-Instruct. Note that the validation loss is only calculated after each epoch and therefore there are only 8 points.

description is shuffled compared to the training data in Fig. 1. Firstly, it is tested with ChatGPT4, which is estimated to have 1.76 trillion parameters (251 times of Llama-2-7B) but it generates irrelevant TCAD files (Fig. 6b). On the other hands, Llama-SJSU-TCAD-base sometimes creates an error-free SSE input file even when it is trained with only 1k data. However, it might also create files with a few mistakes. For example, Fig. 6c shows the responses from Llama-SJSU-TCAD-base for the same instruction which performs much better than ChatGTP4 but fails to understand the meaning of "3.11e18" (it gave 3.11e20). This is the same for Llama2-SJSU-TCAD-chat (Fig. 6d). It also should be noted that the fine-tuned models also perform better than ChatGPT when "What are the differences between a TCAD input file and an English paragraph about a transistor description?" is asked and is able to provide a TCAD input example. Finally, the Llama3-SJSU-TCAD-chat is the most successful model (Fig. 6e). This model can generate the TCAD SSE input file with science notation such as "3.11e18". Especially, it also understands "3.11x10^18" and changes it to "3.11e18".

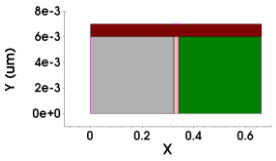
IV. In-context Learning with LLM

In-context learning is a method to aid the LLM models in learning a task by giving one or more examples [6]. However, the model needs to have enough context length (CL). Therefore, we apply it to Llama-3-8B-Instruct (CL = 8k), GPT4o (CL = 128k), and Claude 3.5 Sonnet (CL = 200k). They all have enough context length for the problem we are studying (which is too much for Llama2 with CL = 4k). We gave each model one

a) Instruction: **Generate a TCAD Sentaurus structure input file to create the following** nanowire transistor with a source, a short gate, a long gate, drain, and oxide, as well as three contacts designated for the source, drain, and gate. **The gate contact is notable for its coverage across both the short and long gates. The short gate, on the other hand, is comprised of Silicon, spans 0.005 micrometers,** and is doped with Arsenic at **9e19 cm^-3**. In a similar **vein**, the long gate, also **crafted** from Silicon, has a length of 0.015 micrometers and is doped with Boron at 8.784646123e20 cm^-3. Furthermore, the drain region, **utilizing** Diamond and extending 0.32 micrometers, is doped with Boron at **6.66e20 cm^-3**. Lastly, the gate oxide's thickness is 0.001 micrometers, complementing the nanowire's thickness of 0.006 micrometers. The **construction** includes the source area, which is made of SiGe material, measures 0.32 micrometers in length, and is doped with Phosphorus at a concentration of 3.11e18 cm^-3.

b) Part of ChatGPT 4.0 output: # Sentaurus Structure Definition for a Multi-Gate Nanowire Transistor // # Define materials //Material Si Material SiGe // Material Diamond // Material SiO2 // # Define the device structure // # Geometry dimensions are in micrometers (um) # Source region – SiGe // Region / Source { Material = "SiGe" Doping { Type = "n" Concentration = 3.11e18 } Rectangle { Length = 0.32 Thickness = 0.006 }}

c) Part of the Output Llama-SJSU-TCAD-Base:
(define tox 0.001) // (define Lgs 0.005) // (define Lgl 0.015) // (define Ls 0.32) // (define Ld 0.32)
(define Rl 0.006) // (define Rs 0.006) // (define Xmin 0) // (define Xmax (+ Ls Ltotal Ld))
(sdegeo:create-rectangle (position Xmin Ymin 0) (position XSource Ymax 0) "SiGe" "R.Source")
(sdegeo:create-rectangle (position XSource Ymin 0) (position XGates Rs 0) "Silicon" "R.Short_Gate")
(sdedr:define-constant-profile "Const.Source" Phosphorus **311199862647237219968**)
(sdedr:define-constant-profile "Const.Long_Gate" Boron 878464612339274968576)



d) Part of the Output Llama2-SJSU-TCAD-Chat:
(define tox 0.001) // (define Lgs 0.005) // (define Lgl 0.015) // (define Ls 0.32) // (define Ld 0.32)
(define Rl 0.006) // (define Rs 0.006)// (define Xmin 0) // (define Xmax (+ Ls Ltotal Ld))
(sdegeo:create-rectangle (position Xmin Ymin 0) (position XSource Ymax 0) "SiGe" "R.Source")
(sdegeo:create-rectangle (position XSource Ymin 0) (position XGates Rs 0) "Silicon" "R.Short_Gate")
(sdedr:define-constant-profile "Const.Source" Phosphorus **311948960234138796416**))
(sdedr:define-constant-profile "Const.Long_Gate" Boron 878464612373296374400)

e) Part of the Output Llama3-SJSU-TCAD-Chat:
(define Lgs 0.005)(define Lgl 0.015)(define Ltotal (+ Lgs Lgl))(define Ls 0.32)(define Ld 0.32)(define Rl 0.006)(define Rs 0.006)(define Xmin 0)(define Xmax (+ Ls Ltotal Ld))(define tg 0.001)(define Ymin 0)(define Ymax Rl)(define XSource (+ Xmin Ls))(define XGates (+ XSource Lgs))(define XGatel (+ XGates Lgl))
(sdedr:define-constant-profile "Const.Source"Phosphorus 3.11e18)
(sdedr:define-constant-profile "Const.Short_Gate "Arsenic 9e19)
(sdedr:define-constant-profile "Const.Long_Gate" Boron 8.784646123e20)
(sdedr:define-constant-profile"Const.Drain" Boron 6.66e20)

Fig. 6. Zero-shot with unseen natural language instruction with new statements, shuffled sentences, and uncommonly used English words and phrases (highlighted). b) A part of the output from ChatGPT4. c) A part of the output from Llama-SJSU-TCAD-base with a mistake highlighted and the structure generated. d) A part of the output from Llama-SJSU-TCAD-chat with a mistake highlighted. e) A part of the output from Llama-SJSU-TCAD-Agent, this is correct generation with green highlighted.

"role":"user","content":{TCAD SSE Input File 1st instruction}

"role":"assistant","content":{A TCAD SSE Input File example output }

"role":"user","content":{TCAD SSE Input File 2nd instruction}

Fig. 7. In-context one-shot learning prompt structure of Llama model. The role is blue highlighted, the instruction is orange highlighted, and the example output is green highlighted.

User: "Task: Creating a TCAD Sentaurus structure input file based on instruction. + {TCAD SSE Input File 1st instruction}

Example:{TCAD SSE Input File example output}

Your turn: {TCAD SSE Input File 2nd instruction}"

Fig. 8. In-context one-shot learning prompt structure for ChatGPT and Claude. The role is blue highlighted, the instruction is orange highlighted, and the example output is green highlighted.

example of how to generate a TCAD SSE Input File from human instruction and then asked it to perform another generation with a different human language instruction. This is also called one-shot learning. Note that we will not apply it to fine-tuned models as they have been fine-tuned with the examples. Fig. 7 and Fig. 8 show the setup for in-context one-shot learning for Llama, and ChatGPT and Claude, respectively. Fig. 9 shows the instructions

Complex Instruction: Fabricate a detailed schematic of a nanowire transistor involving a source zone composed of SiGe (length: 0.33 µm, Phosphorus doping: 3.15e18 cm^-3), bifurcated gate sections where the shorter is Silicon-based, 0.028µm in length, Arsenic-doped at 9.1e19 cm^-3, and the longer is Silicon-based, 0.53 µm, Boron-doped at 8.75e20 cm^-3. The drain region utilizes a Diamond substrate, 0.31 µm long, with a Boron doping of 6.75e20 cm^-3. This structure is further refined with a gate oxide layer 0.0068 µm thick and a nanowire thickness of 0.0058 µm. Each segment—source, dual gates, and drain—is equipped with its respective contact. Do not use science notation for number, Ex: for 3.254e17, you should generate a whole number.

Fig. 9. Example of one-shot learning command. The last sentence (red) represents a request to generate input files by converting scientific notation to a non-scientific one.

```
(define Lgs 0.028)//(define Lgl 0.53)//(define Ltotal (+ Lgs Lgl))//(define
Ls 0.33)
(define Ld 0.31)//(define Rl 0.0058)//(define Rs 0.0058)//(define Xmin
0)(define Xmax (+ Ls Ltotal Ld))//(define tg 0.0068)//(define Ymin
0)//(define Ymax Rl)//(define XSource (+ Xmin Ls))//(define XGates (+
XSource Lgs))//(define XGatel (+ XGates Lgl))
(sdegeo:set-default-constant-profile "Const.Source" Phosphorus
3150000000000000000)
(sdegeo:set-default-constant-profile "Const.Short_Gate" Arsenic
9100000000000000000)
(sdegeo:set-default-constant-profile "Const.Long_Gate" Boron
87500000000000000000)
(sdegeo:set-default-constant-profile "Const.Drain" Boron
67500000000000000000)
```

Fig. 10. Part of ChatGPT4o model output after in-context one-shot learning in responds to the request in Fig. 9. It has successfully converted the scientific notation to a non-scientific one.

used. Since numerical representation is a weakness in LLM, we also studied the ability of the LLMs to understand scientific notation by requesting it to convert it into non-scientific notation (last sentence in Fig. 9).

Table I summarizes the results. Llama 2 models do not perform well because they do not have enough context length. ChatGPT 4 also does not perform well despite it being expected to have more than 1 trillion parameters. However, Llama 3, ChatGPT4o, and Claude perform well when there is no scientific notation conversion requirement while they all have fewer parameters than ChatGPT4. The ChatGPT4o is the only model that can understand scientific notation and perform the conversion through in-context one-shot learning for SSE input file generation (Fig. 10).

TABLE I. IN-CONTEXT ONE-SHORT LEARNING PERFORMANCE

Models	Without Scientific Notation Conversion Requirement	With Scientific Notation Conversion Requirement
Llama-2-7B	×	×
Llama-2-7B-chat	×	×
Llama-3-8B-Instruct	✓	×*
Chat-GPT-4	×	×
Chat-GPT-4o	✓	✓
Claude-3.5-Sonnet	✓	×*

×: Bad results. ✓: Good results. *Good results but cannot handle scientific notation.

V. CONCLUSIONS

In conclusion, this research shows a successful application of LLM in TCAD. The fine-tuned Llama models using limited data prove LLMs can be trained to generate accurate TCAD input files. This capability highlights the potential of LLMs to significantly reduce the barrier to entry for engineers and researchers in semiconductor device simulation, making advanced tools like TCAD more accessible and user-friendly.

ACKNOWLEDGMENT

Part of the work was supported by the National Science Foundation under Grant No. 2046220.

REFERENCES

[1] H. Touvron, et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971.

[2] https://openai.com/index/hello-gpt-4o/

[3] https://www.anthropic.com/news/claude-3-5-sonnet

[4] H. Y. Wong, et al., "Enhancement Mode Recessed Gate and Cascode Gate Junctionless Nanowire with Low Leakage and High Drive Current," in IEEE TED, vol. 65, pp. 4004-4008, Sept. 2018.

[5] Y. Yu et al., "Low-Rank Adaptation of Large Language Model Rescoring for Parameter-Efficient Speech Recognition," 2023 IEEE ASRU, Taiwan, 2023, pp. 1-8.

[6] Michael Xie and Sewon Min, (2022, August 5) "How does in-context learning work? A framework for understanding the differences from traditional supervised learning"