

Visualize Me

Explore the quickly updated world

Zining Wang @ Northeastern University

2025-04-04

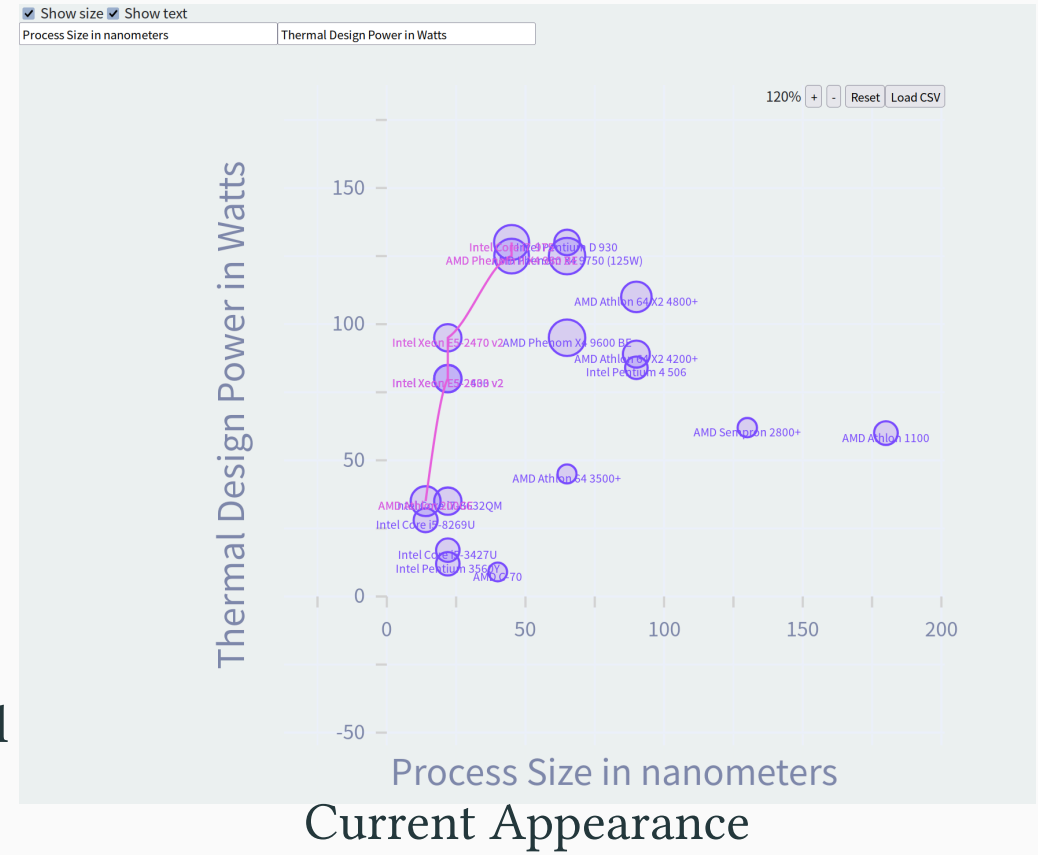
EECE5642 Project Proposal

1. Project Information	2
1.1 Project Information	3
1.2 Motivation	4
1.3 Current Situation	8
2. Develop Process	9
2.1 Developed Part	10
2.2 Live Example	11
2.3 Challenge	12
2.4 Current bugs & Undeveloped Part	13
3. Appendix	14
3.1 project Source Code	15
3.2 slides Source Code	29

1. Project Information

1.1 Project Information

- Team Members & contributions:
 - Zining Wang: Idea, Design, Implementation and Presentation
- Used Language: elm-lang
- Source Code: <https://github.com/wznmickey/visualizeMe>
- Online Demo: <https://wznmickey.github.io/visualizeMe/>
- Data Source:
 - <https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data>
 - <https://llm-stats.com/>



1.2 Motivation

In today's fast-paced technological world, advancements in hardware and artificial intelligence (AI) are occurring at breakneck speed. This creates a dilemma for consumers and professionals who seek to stay up-to-date with the best performing systems without breaking the bank. While tools for benchmarking hardware, particularly CPUs and GPUs, have become increasingly sophisticated, the problem lies in the fact that top-tier performance is not necessarily the most cost-effective solution for everyone.

When selecting hardware, it's important to recognize that performance should not be the only consideration. Several other factors play a crucial role in choosing the right setup for a given application.

1.2 Motivation

- Power Consumption: While higher performance often correlates with higher power requirements, a balance must be struck to avoid excessive energy costs or hardware overheating.
- Price: The most powerful hardware often comes with a premium price tag, but for many users, the best performance is not required for their daily tasks. For example, a developer or gamer may need just enough GPU power to run programs efficiently, but not necessarily the highest-end models available.

1.2 Motivation

- Specific Use Case: Different use cases demand different hardware characteristics.
 - Single-core performance might be essential for applications that rely heavily on sequential processing (e.g., certain older games or single-threaded applications).
 - Multi-core performance is critical for tasks like rendering, scientific computing, and modern gaming where parallel processing is leveraged.
 - For GPUs, some users prioritize high encoding and decoding ability for video production, while others might prefer high memory bandwidth for tasks such as gaming, machine learning, or 3D rendering.

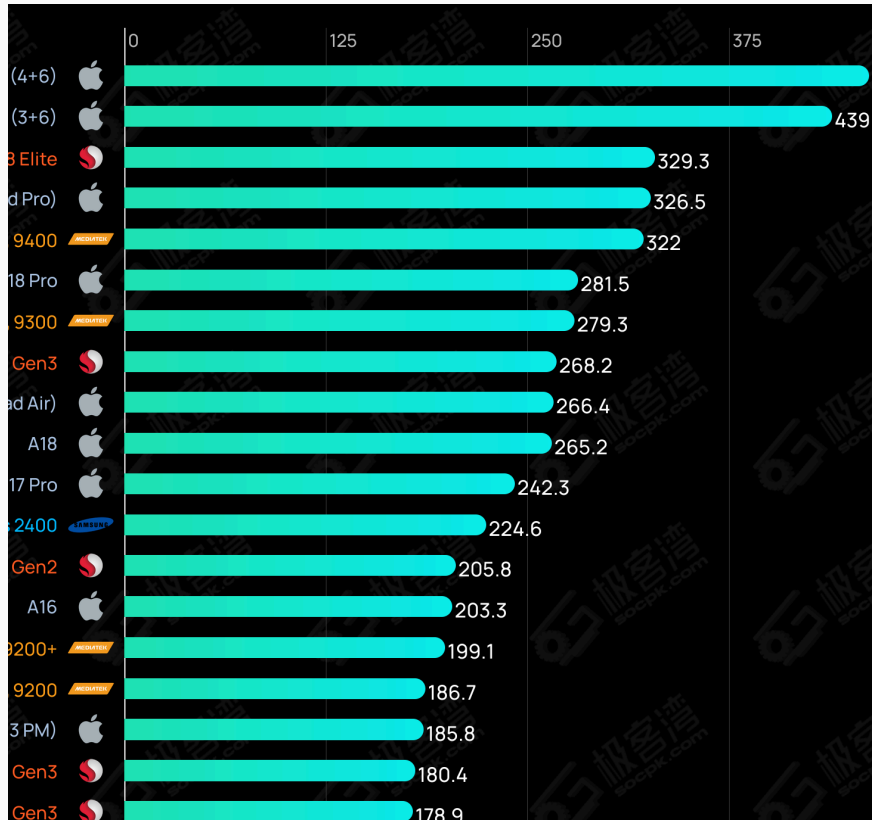
1.2 Motivation

In addition to hardware considerations, there are also other contexts in which performance must be evaluated. For instance, in the realm of Large Language Models (LLMs) and other AI models, the user's needs must be balanced across several dimensions. As LLMs continue to evolve, model size, accuracy, and processing speed are important factors to weigh. Some users may prioritize faster inference times, while others may be more concerned with the accuracy of the results. Additionally, some may be restricted by computational resources, thus requiring smaller models that can still offer competitive performance.

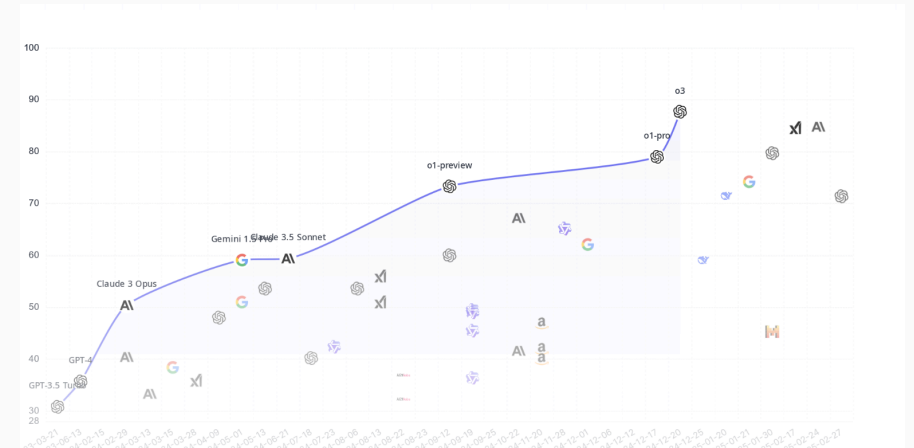
1.3 Current Situation

Hard to filter the desired parameters.

<https://socpk.com/allperf/>



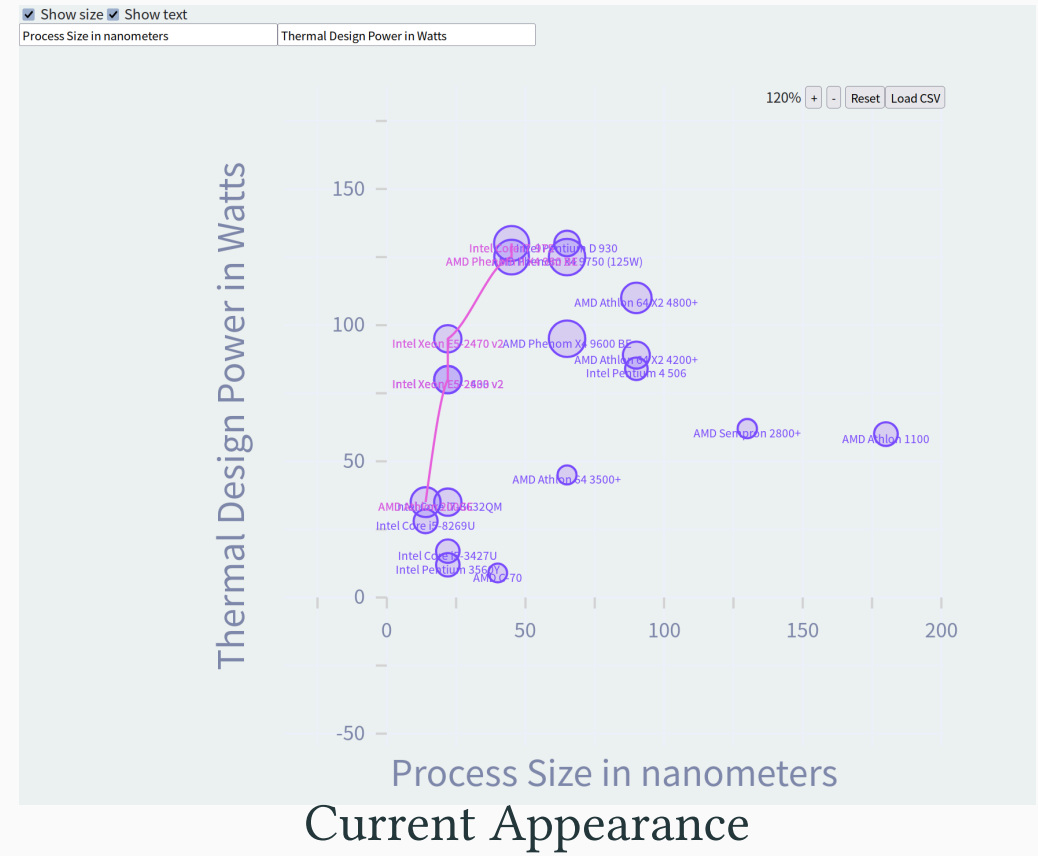
<https://llm-stats.com/>



2. Develop Process

2.1 Developed Part

To make the tool accessible to more people, I want to develop a web-based tool and I use elm-lang which is a functional programming language that could be compiled into a static HTML file with JavaScript containing SVG images which could be easily hosted on GitHub Pages service.



2.2 Live Example

<https://wznmickey.github.io/visualizeMe/>

1. Zoom in and out by pressing the button.
2. Reset the zoom by pressing the button.
3. Zoom in and out by the wheel. (Thanks to the suggestion that using the wheel to zoom in my proposal)
4. Move the plot by clicking the new center.
5. Drag the plot to move it. (Thanks to the suggestion that removing the $+$ line at the center of the screen in my proposal)
6. Switch on the text, size and the pareto line.
7. Upload the data and visualize it.
8. Change the x-axis, y-axis text.

2.3 Challenge

1. Lack of documentation and packages for elm-lang. e.g. csv parsing, key binding
2. Lack of contributor. I work alone in this project.
3. Lack of experience in web frontend development. I am not familiar with tools like CSS in organizing different elements in a page.

2.4 Current bugs & Undeveloped Part

There are some bugs in the current version:

- The text would not disappear when it is out of the screen.
- The size of each point could not resize according to the relative size of the data.
- Text may of x-axis and y-axis may be overlapped by the data point features.
- The layer of the element to capture wheel is lower than that of the element to show the data which causes the wheel event not captured in the center.

To-do:

- Filter different data columns in the csv file.
- When the mouse is moved onto each point, the text of the point should be shown.
- Better design of the layout of the page.

Developed For

computer DIY players

LLM local deployment users & researchers

People who have many options to choose from

<https://wznmickey.github.io/visualizeMe/>

3. Appendix

3.1 project Source Code

Main.elm

```
1  module Main exposing (..)
2
3  import Browser
4  import Zoom exposing (..)
5
6
7  main =
8      Browser.element
9          { init = init
10            , view = view
11            , update = update
12            , subscriptions = \_ -> Sub.none
13            }
14
```

elm

Zoom.elm

```
1  module Zoom exposing (..)
2
3  import Chart as C
4  import Chart.Attributes as CA
5  import Chart.Events as CE
6  import Chart.Item as CI
7  import Chart.Svg as CS
8  import Csv.Decode as Decode exposing (Decoder, column, float, into, pipeline, string)
9  import File exposing (File)
10 import File.Select as Select
11 import Html as H exposing (Html, div)
12 import Html.Attributes as HA exposing (style)
13 import Html.Events as HE
```

elm

3.1 project Source Code

```
14 import Html.Events.Extra.Wheel as Wheel
15 import Svg as S
16 import Svg.Attributes as SA
17 import Task
18
19
20 type alias Point =
21   { x : Float
22     , y : Float
23     , s : Float
24     , w : String
25   }
26
27
28 getPareto : List Point -> List Point
29 getPareto points =
30   let
31     sortedx =
32       List.sortBy .x points
33
34     paretoy =
35       List.foldl
36         (\p acc ->
37           if
38             List.isEmpty acc
39             || (case List.head acc of
40                 Just headPoint ->
41                   headPoint.y < p.y
42
43                 Nothing ->
44                   True
45             )
46         then
47           p :: acc
```

3.1 project Source Code

```
48
49         else
50             acc
51     )
52     []
53     sortedx
54
55     -- Remove duplicates
56     pareto =
57         List.foldl
58         (\p acc ->
59             if
60                 List.isEmpty acc
61                 || (case List.head acc of
62                     Just headPoint ->
63                         headPoint.x /= p.x
64                     Nothing ->
65                         True
66                 )
67             then
68                 p :: acc
69             else
70                 acc
71         )
72     []
73     paretoy
74
75     in
76     List.reverse pareto
77
78
79
80 decoder : Decoder Point
81 decoder =
```

3.1 project Source Code

```
82     into Point
83     |> pipeline (column 1 float)
84     |> pipeline (column 2 float)
85     |> pipeline (column 3 float)
86     |> pipeline (column 0 string)
87
88
89 type alias Model =
90   { center : CS.Point
91     , dragging : Dragging
92     , percentage : Float
93     , data : List Point
94     , showSize : Bool
95     , showText : Bool
96     , showPareto : Bool
97     , textX : String
98     , textY : String
99   }
100
101
102 type Dragging
103   = CouldStillBeClick CS.Point
104   | ForSureDragging CS.Point
105   | None
106
107
108 init : () -> ( Model, Cmd Msg )
109 init _ =
110   ( { center = { x = 0, y = 0 }
111     , dragging = None
112     , percentage = 100
113     , data =
114       [ { x = 65, y = 45, s = 77, w = "AMD Athlon 64 3500+" }
115         , { x = 14, y = 35, s = 192, w = "AMD Athlon 200GE" }
116       ]
117   , Cmd.none
118   )
```

3.1 project Source Code

```
116         , { x = 22, y = 80, s = 160, w = "Intel Xeon E5-2603 v2" }
117         , { x = 45, y = 125, s = 258, w = "AMD Phenom II X4 980 BE" }
118         , { x = 22, y = 95, s = 160, w = "\tIntel Xeon E5-2470 v2" }
119     ]
120     , showSize = True
121     , showText = True
122     , showPareto = False
123     , textX = "Process Size in nanometers"
124     , textY = "Thermal Design Power in Watts"
125 }
126 , Cmd.none
127 )
128
129
130 type Msg
131 = OnMouseMove CS.Point
132 | OnMouseDown CS.Point
133 | OnMouseUp CS.Point CS.Point
134 | OnMouseLeave
135 | OnZoomIn
136 | OnZoomOut
137 | OnZoomReset
138 | FileRequested
139 | FileUpload File
140 | FileLoad String
141 | ToggleShowSize
142 | ToggleShowText
143 | ToggleShowPareto
144 | OnWheelEvent Float
145 | UpdateTextX String
146 | UpdateTextY String
147
148
149 update : Msg -> Model -> ( Model, Cmd Msg )
```

3.1 project Source Code

```
150 update msg model =
151   case msg of
152     OnMouseDown offset ->
153       ( { model | dragging = CouldStillBeClick offset }, Cmd.none )
154
155     OnMouseMove offset ->
156       case model.dragging of
157         CouldStillBeClick prevOffset ->
158           if prevOffset == offset then
159             ( model, Cmd.none )
160
161         else
162           ( { model
163             | center = updateCenter model.center prevOffset offset
164             , dragging = ForSureDragging offset
165           }
166           , Cmd.none
167           )
168
169         ForSureDragging prevOffset ->
170           ( { model
171             | center = updateCenter model.center prevOffset offset
172             , dragging = ForSureDragging offset
173           }
174           , Cmd.none
175           )
176
177     None ->
178       ( model, Cmd.none )
179
180   OnMouseUp offset coord ->
181     case model.dragging of
182       CouldStillBeClick prevOffset ->
183         ( { model | center = coord, dragging = None }, Cmd.none )
```

3.1 project Source Code

```
184
185     ForSureDragging prevOffset ->
186         ( { model
187             | center = updateCenter model.center prevOffset offset
188             , dragging = None
189         }
190         , Cmd.none
191     )
192
193     None ->
194         ( model, Cmd.none )
195
196     OnMouseLeave ->
197         ( { model | dragging = None }, Cmd.none )
198
199     OnZoomIn ->
200         ( { model | percentage = model.percentage + 20 }, Cmd.none )
201
202     OnZoomOut ->
203         ( { model | percentage = max 1 (model.percentage - 20) }, Cmd.none )
204
205     OnZoomReset ->
206         ( { model | percentage = 100, center = { x = 0, y = 0 } }, Cmd.none )
207
208     FileRequested ->
209         ( model
210         , Select.file [ "text/csv" ] FileUpload
211         )
212
213     FileUpload file ->
214         ( model, Task.perform FileLoad (File.toString file) )
215
216     FileLoad str ->
217         ( { model
```

3.1 project Source Code

```
218         | data =
219         case
220             Decode.decodeCsv Decode.NoFieldNames decoder str
221         of
222             Ok points ->
223                 points
224
225             Err err ->
226                 []
227         }
228     , Cmd.none
229 )
230
231 ToggleShowSize ->
232     ( { model | showSize = not model.showSize }, Cmd.none )
233
234 ToggleShowText ->
235     ( { model | showText = not model.showText }, Cmd.none )
236
237 ToggleShowPareto ->
238     ( { model | showPareto = not model.showPareto }, Cmd.none )
239
240 OnWheelEvent delta ->
241     ( { model
242         | percentage =
243             if delta > 0 then
244                 model.percentage + 20
245
246             else
247                 max 1 (model.percentage - 20)
248         }
249     , Cmd.none
250 )
251
```


3.1 project Source Code

```
252     UpdateTextX text ->
253       ( { model | textX = text }, Cmd.none )
254
255     UpdateTextY text ->
256       ( { model | textY = text }, Cmd.none )
257
258
259 updateCenter : CS.Point -> CS.Point -> CS.Point -> CS.Point
260 updateCenter center prevOffset offset =
261   { x = center.x + (prevOffset.x - offset.x)
262   , y = center.y + (prevOffset.y - offset.y)
263   }
264
265
266 view : Model -> Html Msg
267 view model =
268   div []
269     [ div
270       [ style "position" "absolute" ]
271       [ div []
272         [ H.label []
273           [ H.input [ HA.type_ "checkbox", HA.checked model.showSize, HE.onClick ToggleShowSize ] []
274             , H.text " Show size"
275           ]
276         , H.label []
277           [ H.input [ HA.type_ "checkbox", HA.checked model.showText, HE.onClick ToggleShowText ] []
278             , H.text " Show text"
279           ]
280         , H.label []
281           [ H.input [ HA.type_ "checkbox", HA.checked model.showPareto, HE.onClick ToggleShowPareto ] []
282             , H.text " Show pareto line"
283           ]
284       ]
285     , div []
```

3.1 project Source Code

```
286      [ H.input
287        [ HA.value model.textX
288          , HE.onInput UpdateTextX
289        ]
290      []
291    , H.input
292      [ HA.value model.textY
293        , HE.onInput UpdateTextY
294      ]
295    []
296  ]
297 ]
298 , div
299   [ style "width" "100vw", style "height" "100vh", style "overflow" "hidden", Wheel.onWheel chooseZoom ]
300   []
301 , div
302   [ style "position" "absolute"
303     , style "top" "calc(50vh - 40vh)"
304     , style "left" "calc(50vw - 40vh)"
305     , style "width" "80vh"
306     , style "height" "80vh"
307   ]
308   [ C.chart
309     [ CA.height 300
310       , CA.width 300
311       , CA.range [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.x ]
312       , CA.domain [ CA.highest 300 CA.orHigher, CA.zoom model.percentage, CA.centerAt model.center.y ]
313       , CE.onMouseDown OnMouseDown CE.getOffset
314       , CE.onMouseMove OnMouseMove CE.getOffset
315       , CE.on "mouseup" (CE.map2 OnMouseUp CE.getOffset CE.getCoords)
316       , CE.onMouseLeave OnMouseLeave
317     , CA.htmlAttrs
318       [ HA.style "user-select" "none"
319         , HA.style "cursor" "<|
```

3.1 project Source Code

```
320         case model.dragging of
321             CouldStillBeClick _ ->
322                 "grabbing"
323
324             ForSureDragging _ ->
325                 "grabbing"
326
327             None ->
328                 "grab"
329     ]
330 ]
331 [ C.xLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
332 , C.yLabels [ CA.withGrid, CA.amount 5, CA.ints, CA.fontSize 9 ]
333 , C.xTicks [ CA.amount 10, CA.ints ]
334 , C.yTicks [ CA.amount 10, CA.ints ]
335 , C.labelAt CA.middle
336   .min
337   [ CA.moveDown 18 ]
338   [ S.text model.textX ]
339 , C.labelAt .min
340   CA.middle
341   [ CA.moveLeft 18, CA.rotate 90 ]
342   [ S.text model.textY ]
343 , C.series .x
344   [ C.scatter .y [ CA.opacity 0.2, CA.borderWidth 1 ]
345     |> C.variation
346     (\_ d ->
347         [ CA.size
348             (if model.showSize then
349                 d.s * model.percentage / 100 / 10
350
351             else
352                 1
353         )
```

3.1 project Source Code

```
354         , CA.hideOverflow
355     ]
356 )
357 ]
358 model.data
359 , if model.showPareto then
360     C.series .x
361     [ C.interpolated .y [ CA.monotone ] []
362     ]
363     (getPareto model.data)
364
365 else
366     C.series .x
367     [ C.interpolated .y [ CA.monotone ] []
368     ]
369     []
370 , if model.showText then
371     C.eachDot <|
372     \p dot ->
373     [ C.label
374     [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
375     [ S.text (CI.getData dot).w ]
376     (CI.getCenter p dot)
377     ]
378
379 else
380     C.eachDot <|
381     \_ _ ->
382     []
383
384 -- C.eachDot <|
385 -- \p dot ->
386 -- [ C.label
387 -- [ CA.moveDown 4, CA.color (CI.getColor dot), CA.fontSize 5 ]
```

3.1 project Source Code

```
388      --          [ S.text (CI.getData dot).w ]
389      --          (CI.getCenter p dot)
390      --      ]
391      -- , C.withPlane <|
392      --      \p ->
393      --      [ C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.y1 (CA.middle p.y) ]
394      --      , C.line [ CA.color CA.darkGray, CA.dashed [ 6, 6 ], CA.x1 (CA.middle p.x) ]
395      --      ]
396      , C.htmlAt .max
397      .max
398      0
399      0
400      [ HA.style "transform" "translateX(-100%)" ]
401      [ H.span
402          [ HA.style "margin-right" "5px" ]
403          [ H.text (String.fromFloat model.percentage ++ "%") ]
404      , H.button
405          [ HE.onClick OnZoomIn
406          , HA.style "margin-right" "5px"
407          ]
408          [ H.text "+" ]
409      , H.button
410          [ HE.onClick OnZoomOut
411          , HA.style "margin-right" "5px"
412          ]
413          [ H.text "-" ]
414      , H.button
415          [ HE.onClick OnZoomReset ]
416          [ H.text "Reset" ]
417      , H.button [ HE.onClick FileRequested ] [ H.text "Load CSV" ]
418      ]
419      ]
420      ]
421      ]
```

3.1 project Source Code

```
422
423
424 chooseZoom : Wheel.Event -> Msg
425 chooseZoom wheelEvent =
426     case wheelEvent of
427         event ->
428             OnWheelEvent event.deltaY
429
430
431 meta =
432     { category = "Interactivity"
433     , categoryOrder = 5
434     , name = "Zoom"
435     , description = "Add zoom effect."
436     , order = 20
437     }
438
```

3.2 slides Source Code

The typst code generating the slides. `main.typ`

```
1  #import "@preview/touying:0.6.1": *
2  #import "@preview/pinit:0.2.0": *
3  #import themes.metropolis: *
4  #import "@preview/numbly:0.1.0": numbly
5  #import "@preview/codly:1.0.0": *
6
7  #show: codly-init.with()
8  #show: metropolis-theme.with(
9    aspect-ratio: "16-9",
10    footer: self => self.info.institution,
11    config-info(
12      title: [Visualize Me],
13      subtitle: [Explore the quickly updated world],
14      author: [Zining Wang \@ Northeastern University],
15      date: datetime.today(),
16      institution: [EECE5642 Project Proposal ],
17    ),
18  )
19  #set heading(numbering: numbly("{1}.", default: "1.1"))
20  #title-slide()
21  = Outline <touying:hidden>
22
23  #outline(title: none, indent: 1em, depth: 2)
24  = Project Information
25  == Project Information
26  #grid(columns: 2)[
27    - Team Members & contributions:
28      - Zining Wang: Idea, Design, Implementation and Presentation
29
30    - Used Language: elm-lang
31
```

typ

3.2 slides Source Code

```
32 - Source Code: https://github.com/wznmickey/visualizeMe
33
34 - Online Demo: https://wznmickey.github.io/visualizeMe/
35
36 - Data Source:
37   - https://www.kaggle.com/datasets/michaelbryantds/cpu-and-gpu-product-data
38   - https://llm-stats.com/
39 ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
40   Current Appearance]]
41
42 == Motivation
43
44 In today's fast-paced technological world, advancements in hardware and artificial intelligence (AI) are occurring at breakneck speed. This creates a dilemma for consumers and professionals
45 who seek to stay up-to-date with the best performing systems without breaking the bank. While tools for benchmarking hardware, particularly CPUs and GPUs, have become increasingly
46 sophisticated, the problem lies in the fact that top-tier performance is not necessarily the most cost-effective solution for everyone.
47
48 When selecting hardware, it's important to recognize that performance should not be the only consideration. Several other factors play a crucial role in choosing the right setup for a given
49 application.
50 #pagebreak()
51 - Power Consumption: While higher performance often correlates with higher power requirements, a balance must be struck to avoid excessive energy costs or hardware overheating.
52
53 - Price: The most powerful hardware often comes with a premium price tag, but for many users, the best performance is not required for their daily tasks. For example, a developer or gamer may
54 need just enough GPU power to run programs efficiently, but not necessarily the highest-end models available.
55 #pagebreak()
56 - Specific Use Case: Different use cases demand different hardware characteristics.
57   - Single-core performance might be essential for applications that rely heavily on sequential processing (e.g., certain older games or single-threaded applications).
58
59   - Multi-core performance is critical for tasks like rendering, scientific computing, and modern gaming where parallel processing is leveraged.
60
61   - For GPUs, some users prioritize high encoding and decoding ability for video production, while others might prefer high memory bandwidth for tasks such as gaming, machine learning, or 3D
62 rendering.
63 #pagebreak()
64 In addition to hardware considerations, there are also other contexts in which performance must be evaluated. For instance, in the realm of Large Language Models (LLMs) and other AI models,
65 the user's needs must be balanced across several dimensions. As LLMs continue to evolve, model size, accuracy, and processing speed are important factors to weigh. Some users may prioritize
66 faster inference times, while others may be more concerned with the accuracy of the results. Additionally, some may be restricted by computational resources, thus requiring smaller models
67 that can still offer competitive performance.
```


3.2 slides Source Code

```
60
61 == Current Situation
62 Hard to filter the desired parameters.
63 #grid(columns: (1fr, 0.3fr, 1fr))([https://socpk.com/allperf/
64   #image("socpk.png", height: 80%)]][
65   https://llm-stats.com/
66   #image("LLM.png")
67 ]
68
69
70
71 == Develop Process
72
73 == Developed Part
74 #grid(columns: 2)[
75   To make the tool accessible to more people, I want to develop a web-based tool and I use elm-lang which is a functional programming language that could be compiled into a static HTML file
76   with JavaScript containing SVG images which could be easily hosted on GitHub Pages service.
77 ][#grid(columns: 1, align: center)[#image("tempResult.png") ][
78   Current Appearance]]
79
80
81 == Live Example
82
83
84 1. Zoom in and out by pressing the button.
85 2. Reset the zoom by pressing the button.
86 3. Zoom in and out by the wheel. (Thanks to the suggestion that using the wheel to zoom in my proposal)
87 4. Move the plot by clicking the new center.
88 5. Drag the plot to move it. (Thanks to the suggestion that removing the + line at the center of the screen in my proposal)
89 6. Switch on the text, size and the pareto line.
90 7. Upload the data and visualize it.
91 8. Change the x-axis, y-axis text.
92
```

3.2 slides Source Code

```
93 == Challenge
94
95 1. Lack of documentation and packages for elm-lang. e.g. csv parsing, key binding
96 2. Lack of contributor. I work alone in this project.
97 3. Lack of experience in web frontend development. I am not familiar with tools like CSS in organizing different elements in a page.
98
99
100 == Current bugs & Undeveloped Part
101
102 There are some bugs in the current version:
103 - The text would not disappear when it is out of the screen.
104 - The size of each point could not resize according to the relative size of the data.
105 - Text may of x-axis and y-axis may be overlapped by the data point features.
106 - The layer of the element to capture wheel is lower than that of the element to show the data which causes the wheel event not captured in the center.
107
108 To-do:
109 - Filter different data columns in the csv file.
110 - When the mouse is moved onto each point, the text of the point should be shown.
111 - Better design of the layout of the page.
112
113 #focus-slide[
114
115 *Developed For*
116
117 computer DIY players
118
119 LLM local deployment users & researchers
120
121 People who have many options to choose from
122
123 https://wznmickey.github.io/visualizeMe/
124 ]
125
126 == Appendix
```

3.2 slides Source Code

```
127
128 == project Source Code
129 `Main.elm`
130 #{
131   set text(size: 8pt)
132   let x = read("../src/Main.elm")
133   raw(x, block: true, lang: "elm")
134 }
135 `Zoom.elm`
136 #{
137   set text(size: 8pt)
138   let x = read("../src/Zoom.elm")
139   raw(x, block: true, lang: "elm")
140 }
141
142
143 == slides Source Code
144 The typst code generating the slides.
145 `main.typ`
146 #{
147   set text(size: 8pt)
148   let x = read("../main.typ")
149   raw(x, block: true, lang: "typ")
150 }
151
```