Exploring Next-Generation Numbers for Generative Artificial Intelligence

Himeshi De Silva, Nhut-Minh Ho

A*STAR Singapore, National University of Singapore

himeshi_de_silva@a-star.edu.sg, minhhn@comp.nus.edu.sg

Outline

- Background and Motivation
- Qtorch2 & LM Evaluation Harness Integration
- Numerical behavior of recent AI Models
- Experiments and Results
- Discussion and Future work

Background

- Number of parameters in large language models such as the latest GPT can range in the trillions
- Growing movement towards smaller, open-source, open-weight models
- Mixed precision, quantization, low-bit numbers reduce model size
- Qtorch+
- Posits
- Evaluating LLMs

Qtorch2

- Fully compatible with Pytorch 2.4+
- Supports current SOTA models on HuggingFace
- Qtorch2
  - Intercepts tensor operations in Pytorch
  - Dequantizes values to FP32
  - Performs model computations
  - Converts results to FP32

Qtorch2

- LM Evaluation Harness Integration
  - Allows benchmarking on popular academic and industry benchmarks
  - Supports models on HuggingFace and local models
  - Simulate quantization of SOTA models on latest benchmarks

Qtorch2

- Loading models in BFloat16 (work-in-progress)

```
p = bfloat16_posit8_quantize(a,nsize=8,es=1)
print(p.dtype)
print(p)
```

```
torch.bfloat16
tensor([-20.0000, -16.0000, -15.0000, -12.0000, -10.0000,  -7.5000,  -5.0000,
         -2.5000,   0.0000,   2.5000,   5.0000,   7.5000,  10.0000,  12.0000,
         15.0000,  16.0000], dtype=torch.bfloat16)
```

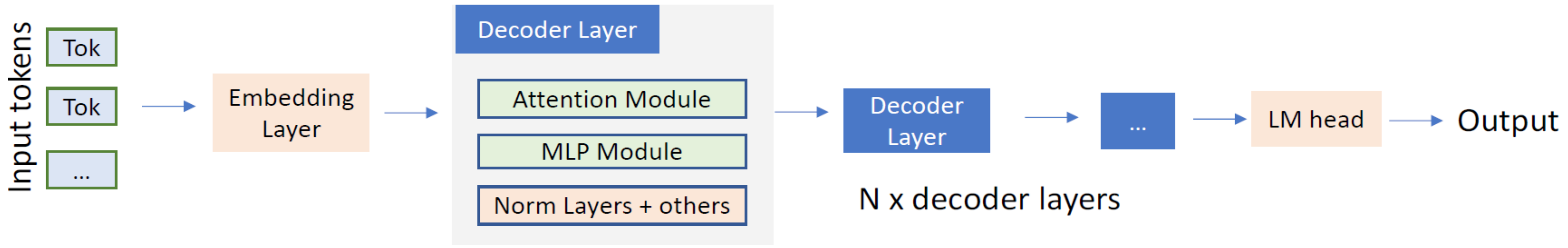
```
p2= bfloat16_posit8_quantize(a,nsize=7,es=1)
print(p2.dtype)
print(p2)
```

```
torch.bfloat16
tensor([-16.0000, -16.0000, -16.0000,  -8.0000,  -6.0000,  -4.0000,  -6.0000,
         -1.5000,   0.0000,   1.5000,   6.0000,   4.0000,   6.0000,   8.0000,
         16.0000,  16.0000], dtype=torch.bfloat16)
```

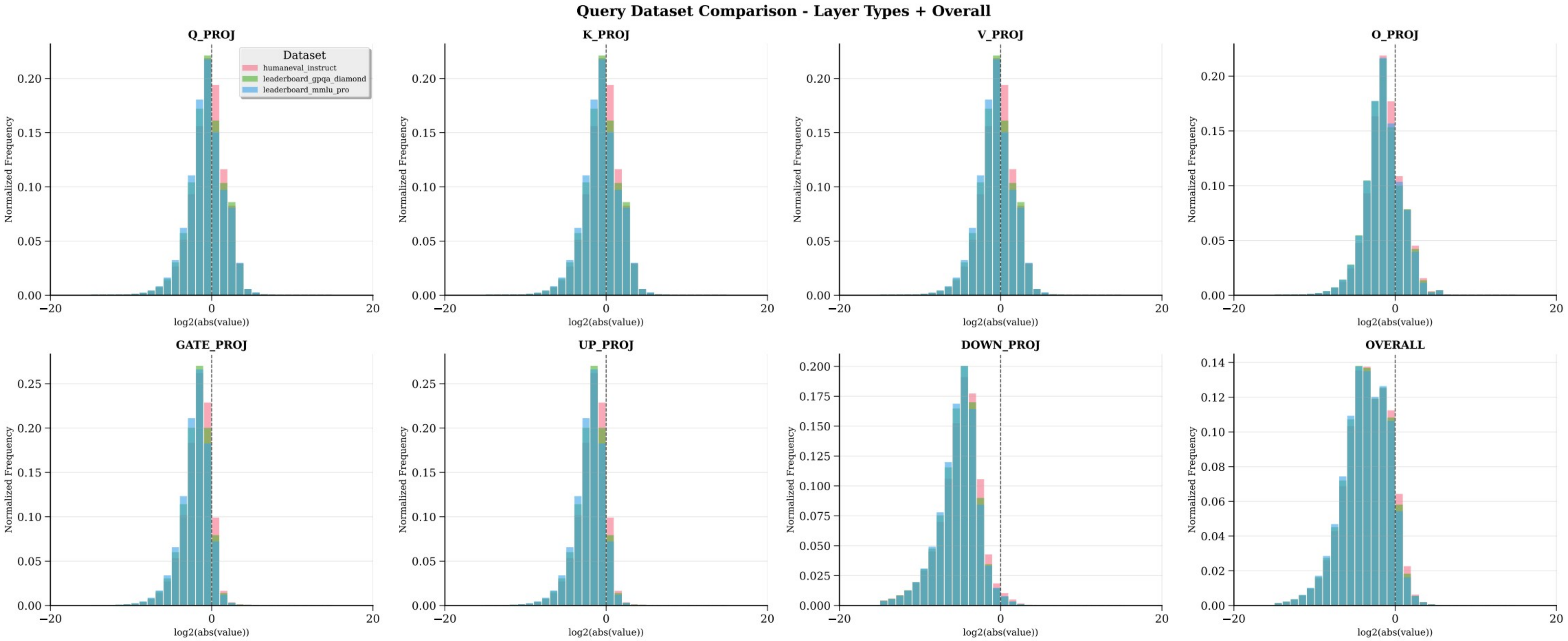
Numerical Characteristics of Recent AI models

LLMs

- Computational cost greater in linear projections used in attention blocks and MLP blocks

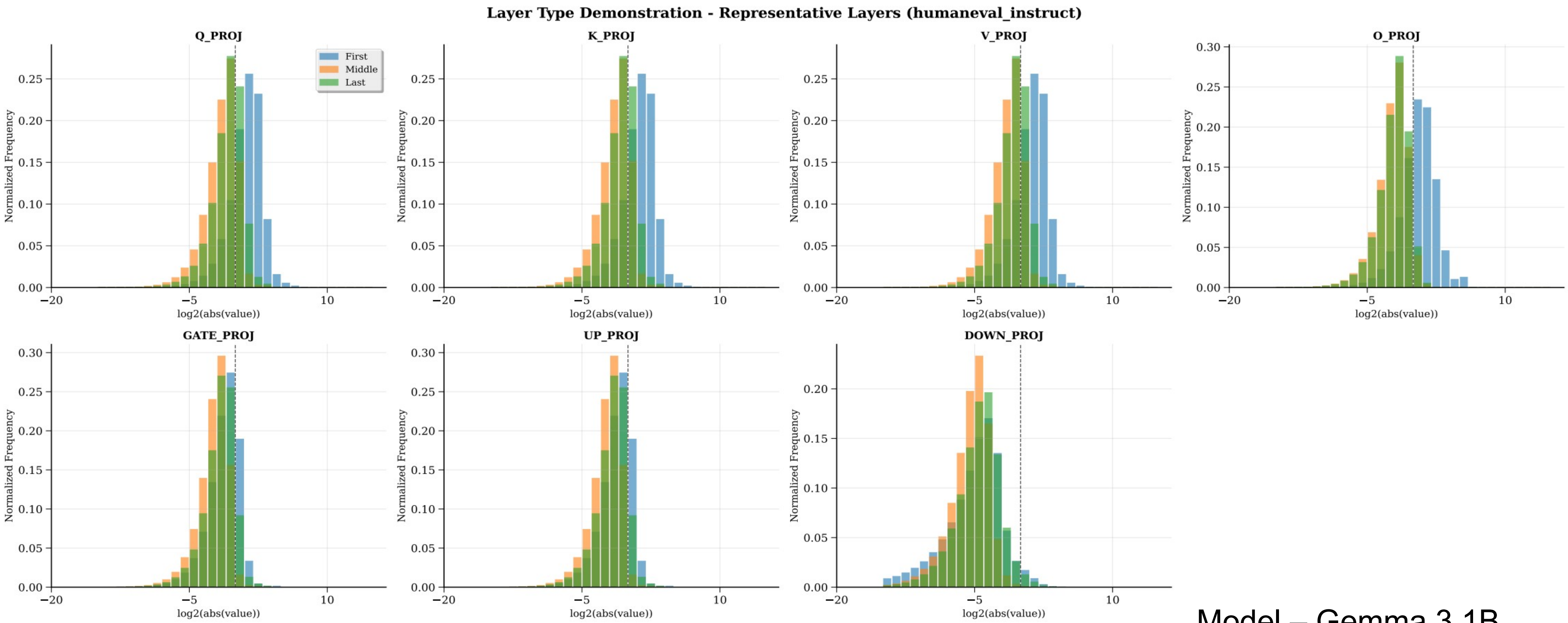


Activations across different inputs



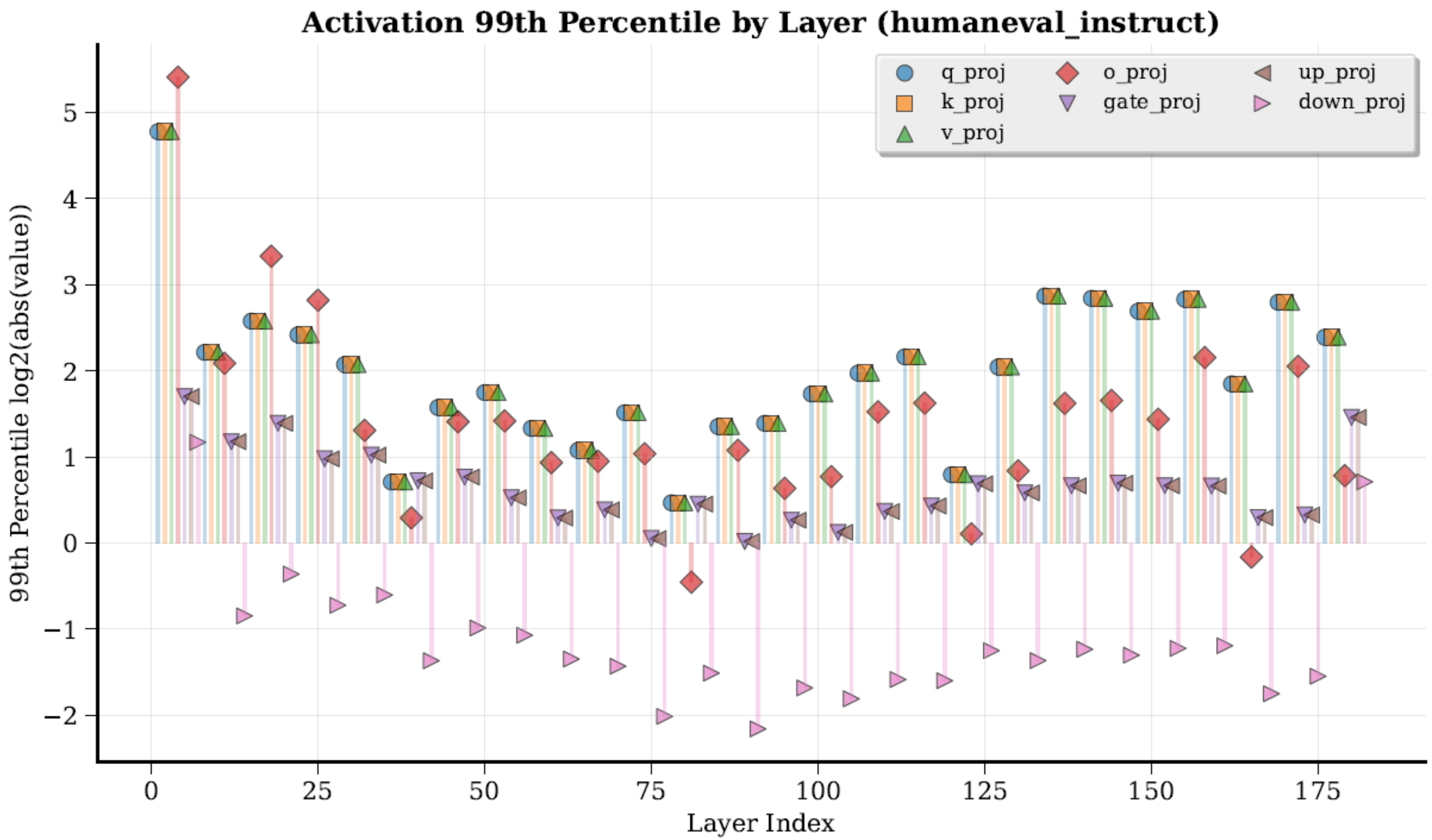
Model – Gemma 3 1B

Activations of different layers of the same type in the same model



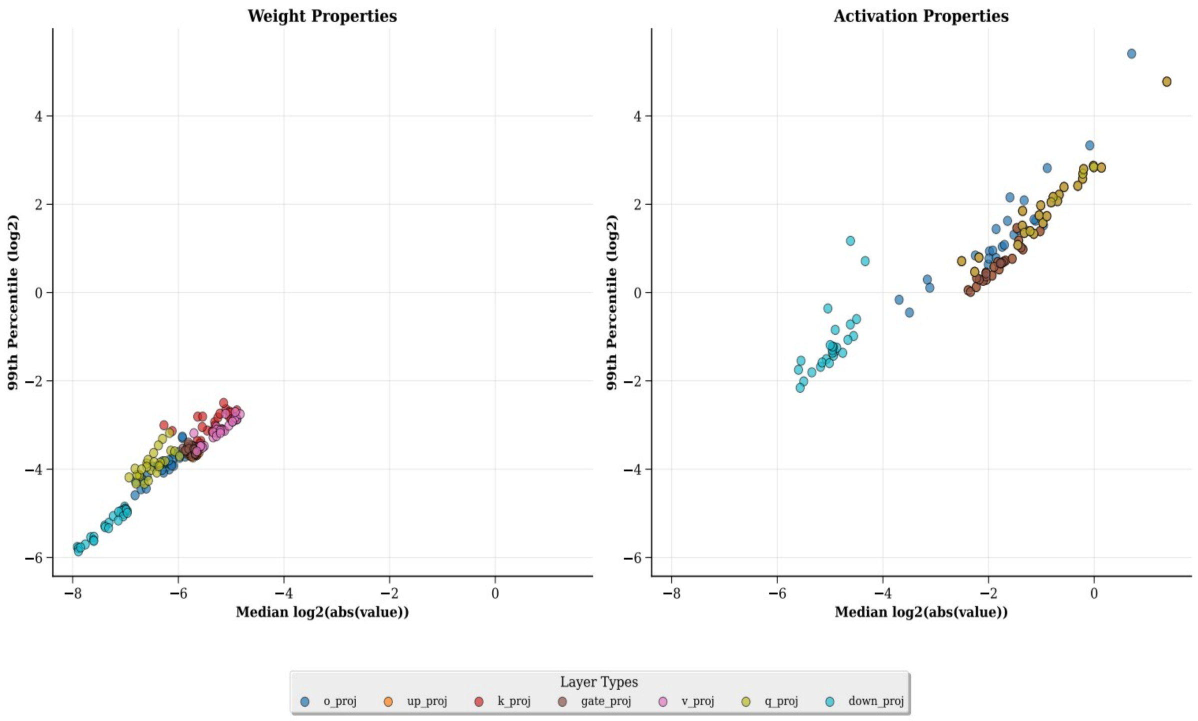
Model – Gemma 3 1B

Activations of different layers of the same type in the same model

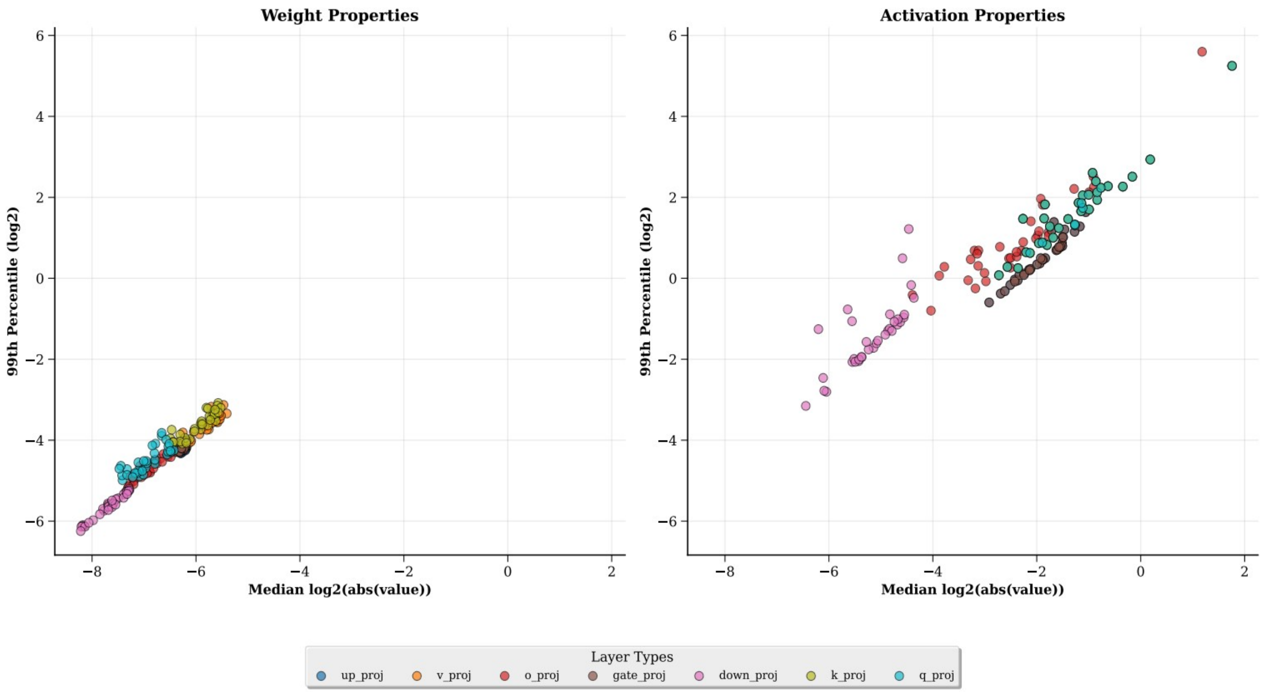


Model – Gemma 3 1B

Weights and activations of models of different sizes in the same family

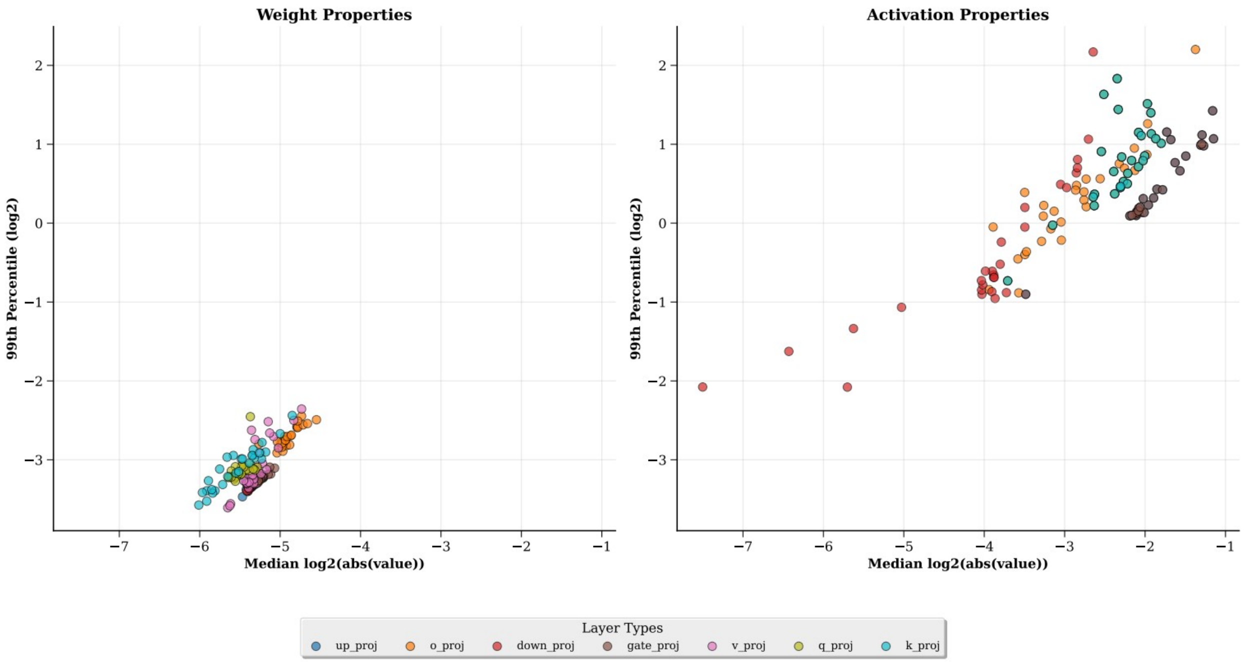


Model – Gemma 3 1B

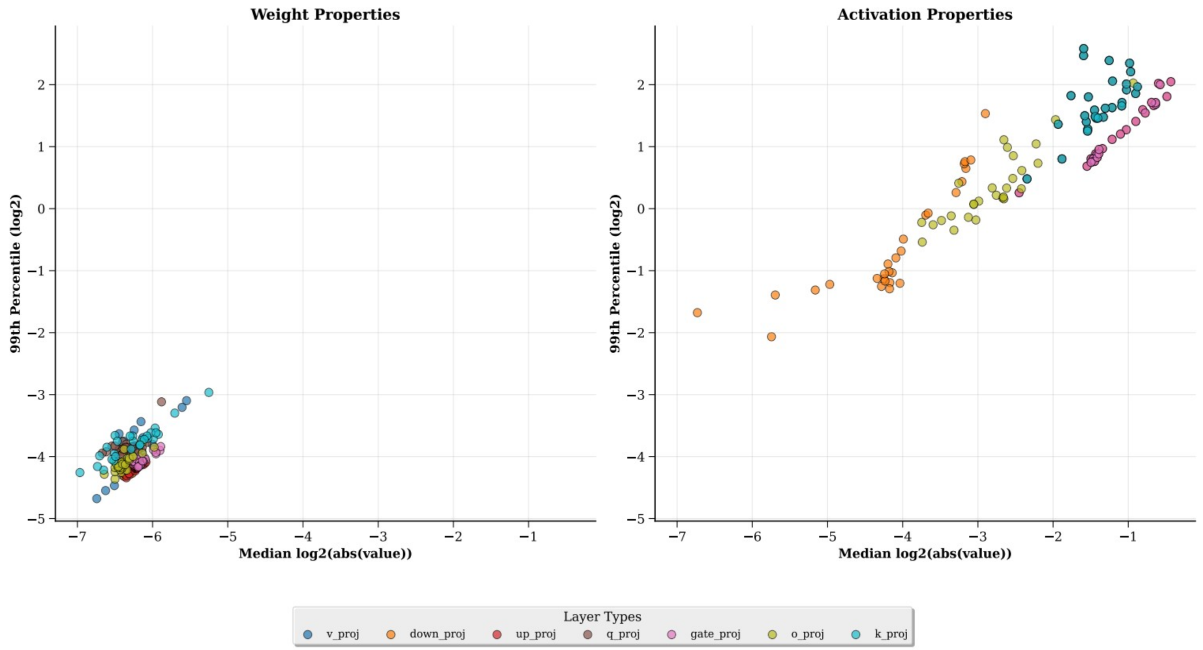


Model – Gemma 3 4B

Weights and activations of models same size of different families



Model – Deepseek R1 1.5B



Model – Qwen 2.5 1.5B

Summary of Observations

- Activations and weights have different numerical characteristics
- Variation of numerical distribution across different benchmarks is small
- There is some variation across layers based on their layer index => Exponent scaling
- Models in the same family display similar numerical behavior

Experiments

1– Standard posits, gradually reduce bitwidth

2 – Apply exponent scaling

- Models – Gemma 1B & 4B, Qwen 2.5 1.5B, Deepseek R1 1.5B
- Benchmarks – ARC Challenge, GPQA Diamond

Results – ARC Challenge

	Original	P(8,1,1)	P(8,1,2)	P(6,1,1)	P(6,1,2)	P(4,1,1)	P(4,1,2)	P(3,1,1)	P(3,1,2)
Gemma_3_4B	53.3	**51.9**	52.4	29.3	28.6	20.7	21.5	22.5	23.4
Gemma_3_1B	35.2	35.6	**36.3**	21.3	21.2	21.0	22.6	21.8	22.1
Qwen_2.5_1.5B	39.2	**32.0**	29.4	19.8	28.8	22.4	21.2	22.7	23.7
DeepSeek_R1_1.5B	32.3	26.1	**29.9**	20.1	19.5	22.8	23.3	25.3	21.7

P(8,1,1) => 8-bit posits, weight exponent = 1, activation exponent = 1

Results – GPQA Diamond

	Original	P(8,1,1)	P(8,1,2)	P(6,1,1)	P(6,1,2)	P(4,1,1)	P(4,1,2)	P(3,1,1)	P(3,1,2)
Gemma3_4B	35.4	**31.8**	29.8	30.3	26.3	18.2	27.3	23.7	26.8
Gemma3_1B	24.2	24.2	23.2	24.2	**30.3**	25.8	24.7	19.2	23.2
Qwen_2.5_1.5B	24.7	23.2	24.2	21.7	24.2	22.2	**26.8**	20.2	18.7
DeepSeek_R1_1.5B	31.3	**29.8**	26.8	26.8	**29.8**	24.7	20.2	28.3	21.7

Results – ARC Challenge with scaling

	P(8,1,1)-(5,1)	P(8,1,1)-(5,2)	P(8,1,2)-(5,1)	P(8,1,2)-(5,2)
Gemma_3_4B	52.4	51.5	53.2	**53.3**
Gemma_3_1B	35.3	34.5	35.1	**35.6**
Qwen_2.5_1.5B	32.7	**38.4**	30.6	28.8
DeepSeek_R1_1.5B	24.6	23.5	**30.2**	27.0

Results – GPQA Diamond with scaling

	P(8,1,1)-(5,1)	P(8,1,1)-(5,2)	P(8,1,2)-(5,1)	P(8,1,2)-(5,2)
Gemma3_4B	33.3	30.3	**36.4**	34.3
Gemma3_1B	22.7	**27.8**	24.7	25.3
Qwen_2.5_1.5B	24.2	22.2	28.3	**30.3**
DeepSeek_R1_1.5B	26.8	**28.8**	27.8	27.8

P(8,1,1)-(5,1) => 8-bit posits, weight exponent = 1, activation exponent = 1, weight bias = 5, activation bias = 1

Results – HumanEval

	Original	P(8,1,1)	P(8,1,2)	P(6,1,1)	P(6,1,2)	P(4,1,1)	P(4,1,2)
Qwen_2.5_coder_1.5B	63.4	16.5	1.8	0.0	6.1	0.0	0.0
Gemma_3_4B	66.5	68.3	67.7	4.9	0.6	0.0	0.0

Without exponent bias

	P(8,1,1)-(5,0)	P(8,1,1)-(5,2)	P(8,1,2)-(5,0)	P(8,1,2)-(5,2)
Qwen_2.5_coder_1.5B	14.0	60.4	1.2	6.1
Gemma_3_4B	65.2	67.1	67.7	65.9

With exponent bias

Discussion & Future Work

- Enhancements to simulation of next-gen numbers with Qtorch2, integration with LM evaluation harness
- Selected posit configurations can be generalized with exponent bias
- Further bitwidth reduction with layer-wise exponent bias
- Test newer variations of posit and other number representations