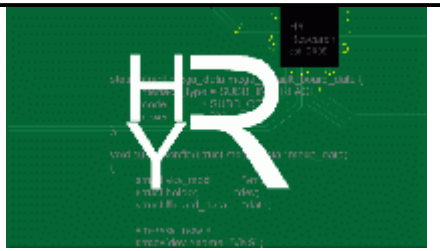
Leveraging LLMs on Embedded Devices

SCALE23X

March 2026

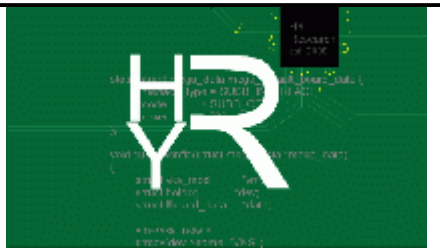
Rev 1.3



http://www.hy-research.com/

Agenda

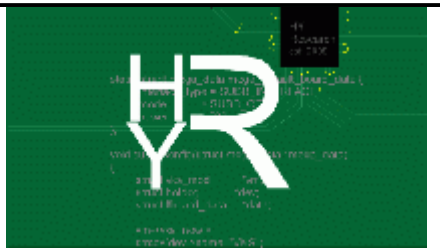
- Introduction
- What/Why LLMs?
- LLM and Machine Learning
- Embedded devices Limitations
- Embedded devices and LLMs
- Fitting an LLM
- LLM limitations
- LLM Model selection
- Look at some data
- Testability
- Security
- Alternatives
- Conclusion



http://www.hy-research.com/

Introduction

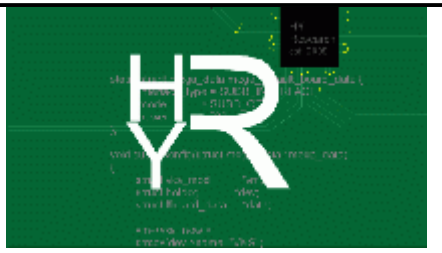
- Sharing things learned as an embedded dev
- Assumptions
  - Basic understanding of machine learning
- Focus on embedded along with its limitations
- Inference only
- Local execution
  - No cloud backend
- Focus on embedded Linux



http://www.hy-research.com/

Why/What is an LLM?

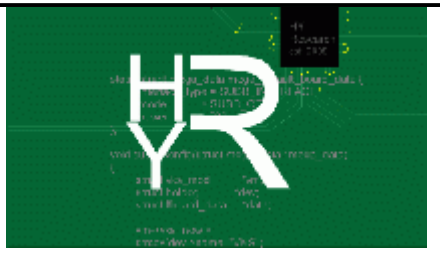
- Large Language Model
- Trained on a very large dataset
  - Text and even include media like static or video images and audio
  - “Understands”/can interact using **human languages**
- Training allows it to digest/match complex patterns and relate them to concepts
  - Larger dataset allows more matching (“knowledge”)
- LLMs work on **tokens**
  - **All** input and output are tokens



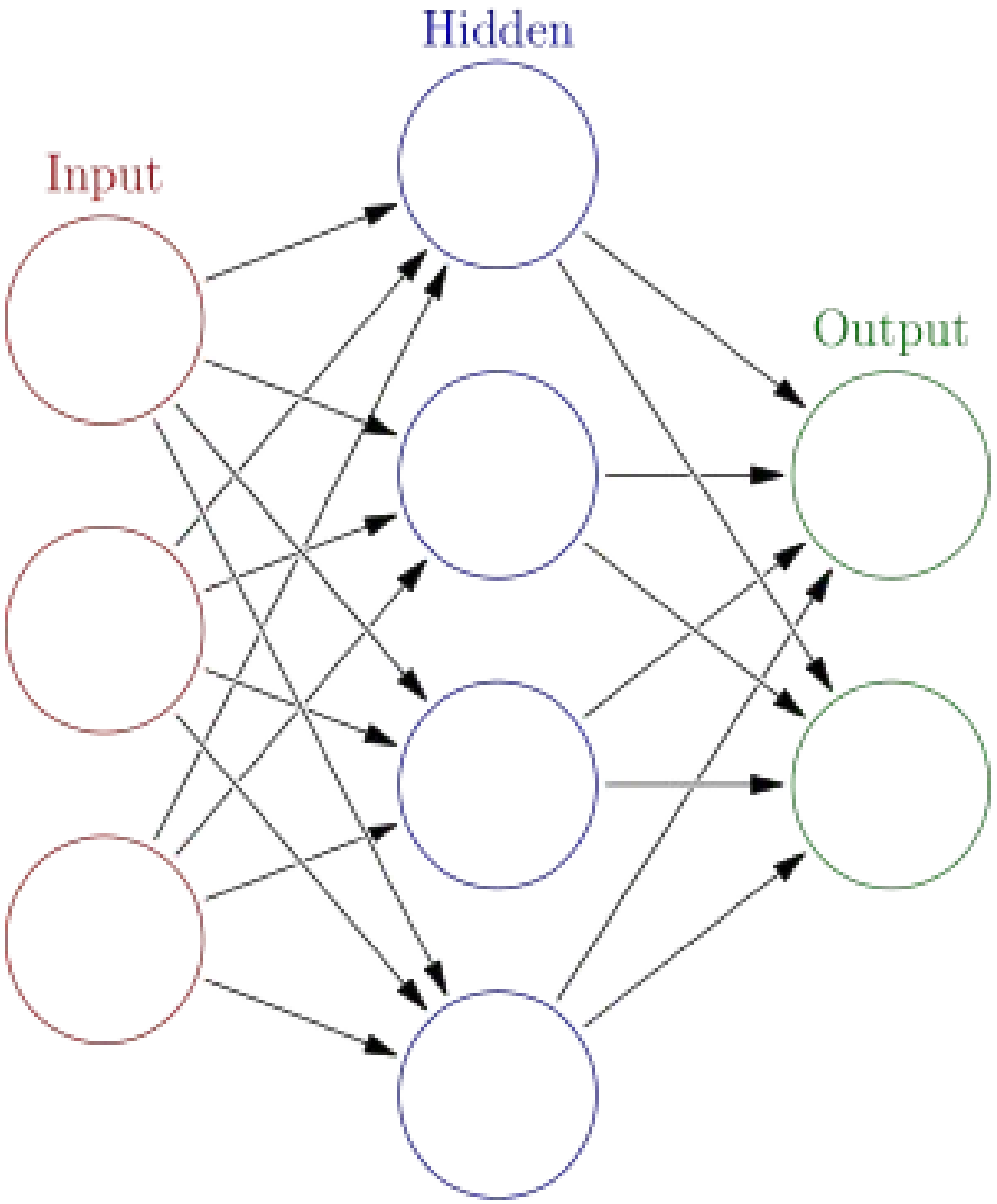
http://www.hy-research.com/

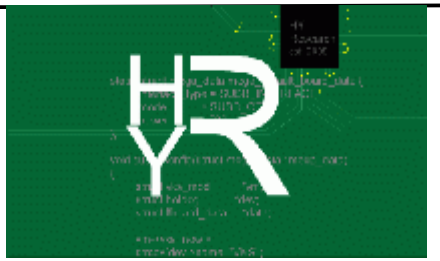
LLM and Machine learning

- Most popular are the transformer model
  - Transform an input into an output
    - “Translations”
- More complex but same basic idea
  - Training takes “true”/”data”/”knowledge” to tune weights that generates an output that best matches the expected output.
- Each neuron - mapping(Input * Weight + offset)
  - Explicit randomness factor, temperature
  - Multiply/add limited dependencies
    - Can run in parallel



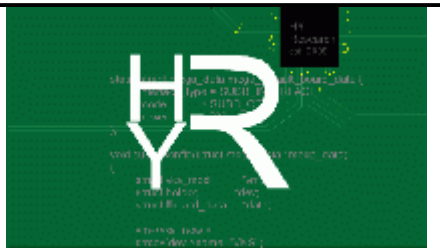
Neural Network





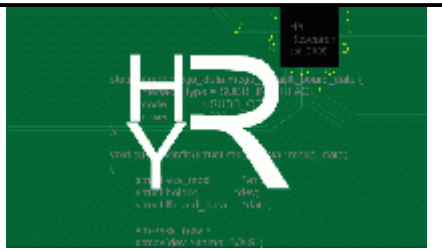
LLM helpers

- LLMs often need a helper model for media beyond plain text
  - Images need a mmproj helper model to encode images
  - Turns the media into tokens
  - Usually smaller the main model
  - Specific to the main model/model family



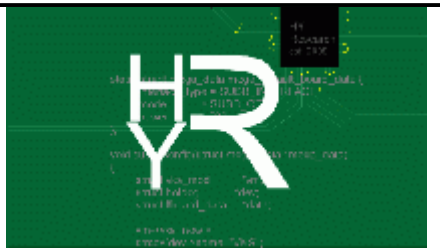
Embedded Device Limitations

- Resources are finite/dictated by application
  - Electrical power limited – Battery, thermal, etc.
  - Memory and Processing – physical size, power
  - Connectivity limited/nonexistent
- Application may dictate real time processing
- Other processing may be required
  - Sensor servicing
  - Out processing
  - UI handling



Embedded device and LLM

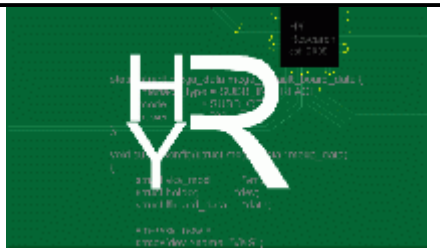
- LLMs can offer unique features
- Vision/hearing like processing of optical/audio data
- Simplified/more natural interaction
  - Voice interactions
  - Natural language
- Complex data analysis
- LLMs at different points (can be interleaved or chained)
  - Input
  - Processing
  - Output



Fitting an LLM

- LLMs are HUGE
  - Both in memory and processing terms
  - Can exceed even common desktop
  - IO is significant
- Common resource requirements
  - GPU for processing
  - Multi Gigabyte memory
  - Memory bandwidth/IO bandwidth

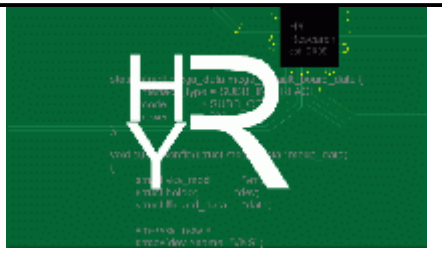
https://commons.wikimedia.org/wiki/File:Tesla-NVIDIA_GPU_cluster_(3706444821).jpg



http://www.hy-research.com/

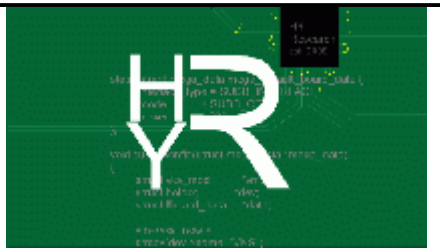
Fitting an LLM - approaches

- Multiple SoCs – dedicate an SoC to run the LLM and aggressively apply PM techniques ($$$)
- Add a GPU – include a desktop GPU in your device and apply PM techniques ($$$$)
- CPU core choices – hardware multipliers, SIMD instructions (NEON, MMX, SSD, etc)
- Hardware Accelerators – NPUs, dedicated acceleration chips ($$)
  - Vendor lock in/possible supply chain cautions



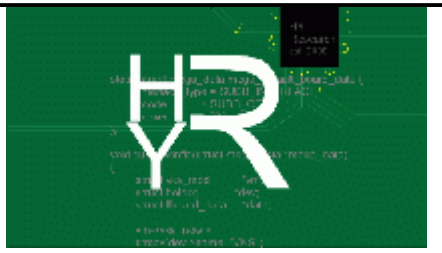
Fitting an LLM - (con't)

- SoC with combinations of above
  - Nvidia Tegra based offering ($$) includes a Nvidia GPU (CUDA capable) in SoC for embedded use
- Careful selection of models
  - Model size – smaller number of parameters
  - Model design
- Selective usage of the LLM
  - Only use the LLM when really needed
  - Rework things to lower the realtime demand for the pieces that require an LLM



Fitting an LLM - approaches

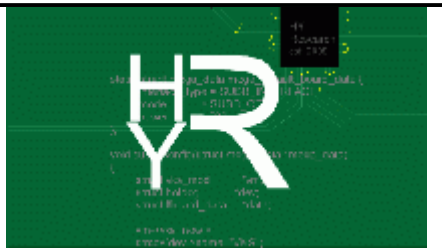
- Memory limited vs Processing limited
- Model selection can have a huge effect
  - Model architecture
  - Model size
- Inference Implementation
  - Implementation (Python vs a pure C implementation)
  - Model quantization (number of bits to represent numbers)
  - Numeric (float vs int)
- Inference method (acceleration)



http://www.hy-research.com/

LLM Limitations

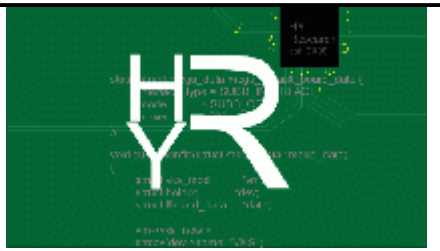
- Output may not be deterministic
- Slight changes in the input may alter output dramatically
- Output may be **completely wrong** (Hallucinations)
  - Small model limitations
  - Gets better with larger models
  - **Does not completely go away**
  - Output may make accurate sounding claims that are completely wrong



http://www.hy-research.com/

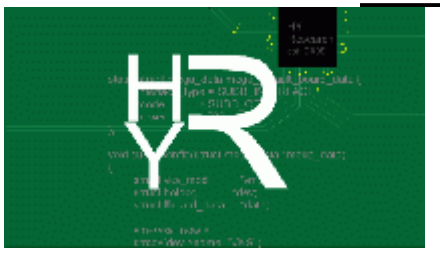
LLM Limitations (con't)

- Output accuracy range of high 90's% so there WILL be wrong outputs (best case)
  - Increasing training helps but will not hit 100%
  - Application specific training can help
  - May not be suitable for critical use cases and such usage may be expressly prohibited by model license terms
- Accuracy factors
  - Quantization
  - Model size (parameters)
  - Numerics
  - Training data
  - Implementation



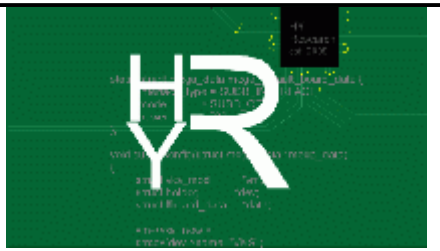
LLM Model selection

- Do you plan to train or attempt to use an open weights model (check license!)?
  - Training can be fine tuning for application specific data OR training from ground up. Need big iron
  - Models available for fine tuning
- Older models may require less hardware/simpler to train
  - PyTorch + Nvidia GPU (with lots of VRAM) is common for many models
  - Opensource GPT2 implementation (including training) in plain C possibly with just desktop CPU
- More parameters is more accurate but parameter size is proportional to memory usage



LLM Model implementation

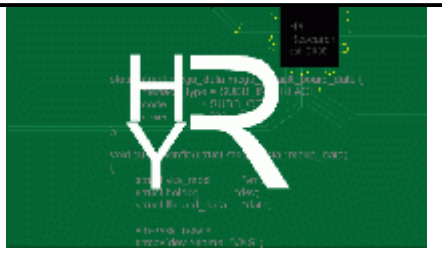
- Quantization (rounding)
  - Number of bits used to represent the weights
  - Range from 2bits up to a full 32bit, fewer bits generally lowers accuracy but not linearly
  - Flash/image requirements
- Numerics
  - Many models are trained using 32bit floating point values for constants/weights
  - Inference can **sometimes** be accelerated by converting them to integers
  - Size of floats f16 vs f32
- Inference
  - Implementation matters. Research/training often use implementations like PyTorch
  - C/C++ implementations may be available



http://www.hy-research.com/

LLM Naming conventions

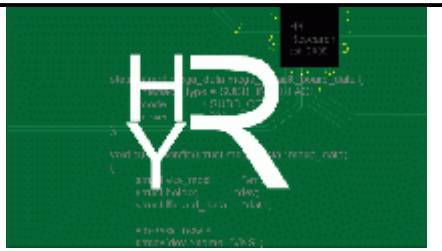
- Not standardized but commonly used are:
  - xB is the “size” or the number of parameters used to train the model. This is the rough number of training data in billion
  - qn is the number of bits the constants been quantized to
  - fXX typically refers to the model using floating point for the weights of that size. i.e. f16 is a 16bit float
- Example:
  - SmolVLM2-2.2B-Instruct-f16.gguf



http://www.hy-research.com/

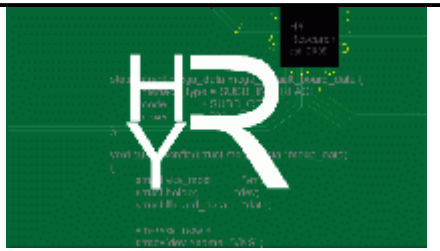
Common open models

- Llama family from Meta
- Gemma family from Google
- Mistral family from Mistral
- Qwen family from Alibaba
- Many open models are available on https://huggingface.co/



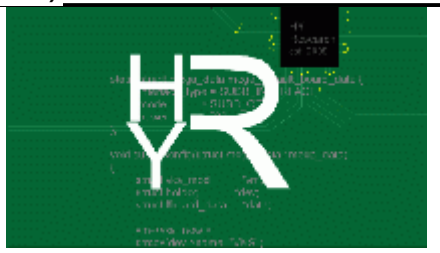
Inference Implementation

- llama.cpp (https://github.com/ggml-org/llama.cpp)
  - C only inference implementation supporting many models
  - Minimal external dependencies – primary dependencies are for acceleration
    - SIMD support on x86 and ARM (maybe other, active development)
    - OpenCL/CUDA, metal (Mac hardware) support
    - No NPU/accelerator support out of the box
    - BLAS libraries
  - Supports acting as a chat endpoint and includes CLI tools to test
  - Supports media (audio, images)
  - Uses .gguf file for model weights and description
    - May need to use tool to convert to the gguf format
- Other C only implementations
  - GPT2 - https://github.com/shreydan/VisionGPT2



Real Data

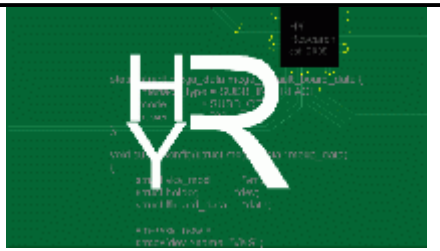
- Use an LLM to annotate a set of images
  - Hypothetical life recorder device – periodic snapshot to journal
  - Trade off – reduce mealtime demand compared to describing live video
- Configuration:
  - Device will be setup as a chat end point
  - Use llama.cpp (https://github.com/ggml-org/llama.cpp)
    - No acceleration
  - Measurements will be from another networked device
- Models used:
  - SmolVLM2 (https://huggingface.co/HuggingFaceTB/SmolVLM2-2.2B-Instruct)
    - 2.2B parameters
  - MiniCPM (https://huggingface.co/openbmb/MiniCPM-V-4_5-gguf/)
    - 8B parameters
- Goals: Measure and evaluate different options
- Prompt: "Concisely describe the image in detail."



Hardware & Data

1) RPi 5
   - Cortex-A76 x 4 @ 2.4GHz 512K L2/core; 2M L3 shared
   - 16G LPDDR4X-4267 (533MB/s, 4267 T/s)
2) Orange Pi5 Ultra
   - Cortex-A76 x 4 @ 2.6GHz 512K L2/core;
   - Cortex-A55 x 4 @ 1.8GHz 128K L2/core; 3M L3 shared w/ 8 cores
   - 16G LPDDR5 (800MB/s, 6400 T/s)

- Both hardware setup have fans for cooling
- Data is mostly 12Mpixel JPEG resized to 1.2M before sent to LLM



http://www.hy-research.com/

Data

RPi5

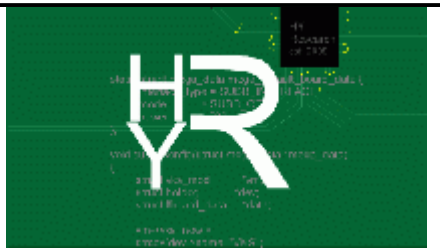
			Time	Watt	Per Img	Notes
2.4GHz	MiniCPM	Q2/f16	4438		57.63	OpenBLAS
2.4GHz	MiniCPM	Q2/f16	3240		42.08	
2.4GHz	MiniCPM	Q4/f16	3931		51.06	
2.4GHz	MiniCPM	Q8/f16	6080		78.96	
2.4GHz	SmolVLM2	Q8/Q8	5189		67.40	
2.4GHz	SmolVLM2	Q4/Q8	3016		39.17	
2.4GHz	SmolVLM2	Q4/f16	4273	12	55.49	

Up to 75C

OrangePI5 Ultra

1.2GHzx8	MiniCPM	Q4/f16	5530		71.81	
1.4GHzx8	MiniCPM	Q4/f16	4712		61.19	
1.4x4,1.6x4	MiniCPM	Q4/f16	4494		58.37	
1.4x4,1.8x4	MiniCPM	Q4/f16	4299		55.83	
2.4x4,1.8x4	MiniCPM	Q4/f16	3662		47.55	
2.4x4,1.8x4	MiniCPM	Q2/f16	4637		60.22	
2.4x4,1.8x4	MiniCPM	Q4+XS/f	3404		44.21	
2.4x4,1.8x4	MiniCPM	Q8/f16	5082		66.00	
2.4x4,1.8x4	SmolVLM2	Q8/Q8	5792		75.22	
2.4x4,1.8x4	SmolVLM2	Q4/Q8	4445	11	57.72	
2.4x4	SmolVLM2	Q4/Q8	3587	11	46.58	Big Only
2.4x4	SmolVLM2	Q4/f16	3432		44.57	Big Only

Up to 75C



http://www.hy-research.com/

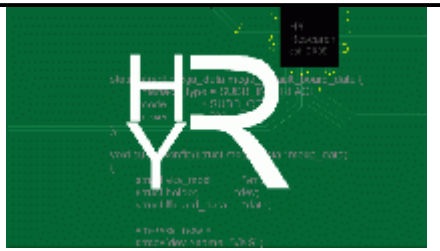
Observations

- Changing model OR quantization can be significant

RPi5		
SmolVLM2	Q8/Q8	5189
MiniCPM	Q8/f16	6080
SmolVLM2	Q4/Q8	3016
MiniCPM	Q4/f16	3931

- Mmproj quantization *can* impact speed

				Time Per	
RPi	2.4GHz	SmolVLM2	Q4/Q8	3016	39.17
RPi	2.4GHz	SmolVLM2	Q4/f16	4273	55.49
Opi	2.4x4	SmolVLM2	Q4/Q8	3587	46.58
Opi	2.4x4	SmolVLM2	Q4/f16	3432	44.57



http://www.hy-research.com/

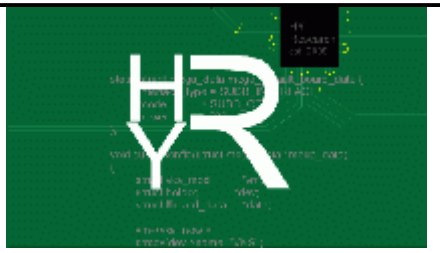
Observations

- OpenBLAS may not help!

MiniCPM	Q2/f16	4438	57.63 OpenBLAS
MiniCPM	Q2/f16	3240	42.08

- Hardware vs Quantization:
  - Fewer bits **can** be faster but not always (yellow vs blue)
  - Floating **can** be faster (but bigger) (orange)

Rpi5			
MiniCPM	Q2/f16	3240	42.08
MiniCPM	Q4/f16	3931	51.06
MiniCPM	Q8/f16	6080	78.96
OrangePi5			
MiniCPM	Q2/f16	4637	60.22
MiniCPM	Q4/f16	3662	47.55
MiniCPM	Q4+XS/f	3404	44.21
MiniCPM	Q8/f16	5082	66.00

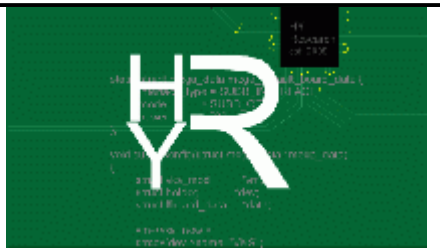


http://www.hy-research.com/

Observations

- Floating vs Quantized
  - Q8 faster then f16 on RPi5 but opposite on OPi5 (for mmproj)
- More cores is not better!

RPi5			
SmolVLM2	Q4/Q8	3016	39.17
SmolVLM2	Q4/f16	4273	55.49
OrangePi5			
SmolVLM2	Q4/Q8	4445	57.72 8 cores
SmolVLM2	Q4/Q8	3587	46.58 4 cores
SmolVLM2	Q4/f16	3432	44.57 4 cores



http://www.hy-research.com/

Inference Result 1

		sku	price	mode	Notes
MiniCPM	Q8/f16	X	X		Fanny Bell Passage Model?
MiniCPM	Q4/f16	X	X	X	Claims of bar code is wrong
MiniCPM	IQ4/f16	X	X	X	
MiniCPM	Q4/f16		X		
MiniCPM	Q2/f16		X		
SmolVLM2	Q8/Q8	X	X	X	Claims of s/n is wrong; Pictu
SmolVLM2	Q4/f16		X		Bar code scanner??
SmolVLM2	Q4/Q8		X		Claims of bar code scanner?'

- Q8 is not better
- Price is extracted consistently
- Obvious hallucination

- Note: SmolVLM2 is 2.2B vs 8B on MiniCPM

Inference Result 2

		Yel F	Yel C	Cactu	Black	Back	Notes
MiniCPM	Q8/f16	X	X	X			
MiniCPM	Q4/f16	X	X	X	X	X	
MiniCPM	IQ4/f16	X	X	X	X		
MiniCPM	Q2/f16	X	??	??	X	??	Spikely Leaf?
SmolVLM2	Q8/Q8	X	??	??		??	Green Leaf?
SmolVLM2	Q4/Q8	X	??	??		??	Claims of blurr
SmolVLM2	Q4/f16	X	??	??		??	High angle obs

- Q8 is not always best
- Not always consistent between runs
- Hallucinations at lower Q2
- Hallucinations SmolVLM2

- Note: SmolVLM2 is 2.2B vs on 8B MiniCPM

http://www.hy-research.com/

Inference 3

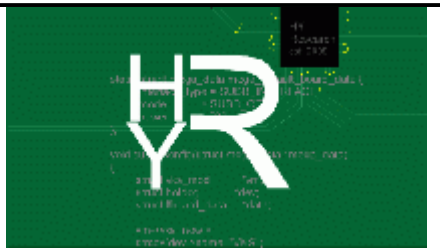
		Summari	Sign Text	
MiniCPM	Q8/f16		X	Claims additional info
MiniCPM	Q4/f16	X	X	Claims additional info
MiniCPM	IQ4/f16		X	Hand Painted? “US 97”? “Rusc
MiniCPM	Q2/f16		X	“9.6 miles from Hwy 97”
SmolVLM2	Q8/Q8		X	Deer Creek Trail?
SmolVLM2	Q4/Q8	X		
SmolVLM2	Q4/f16	X		
SmolVLM2	Q4/Q8		X	“”D. S. BERKELEY & RANDAL

- Q8 isn’t always better (both models)
- Hallucinations

- Note: SmolVLM2 is 2.2B vs on 8B MiniCPM

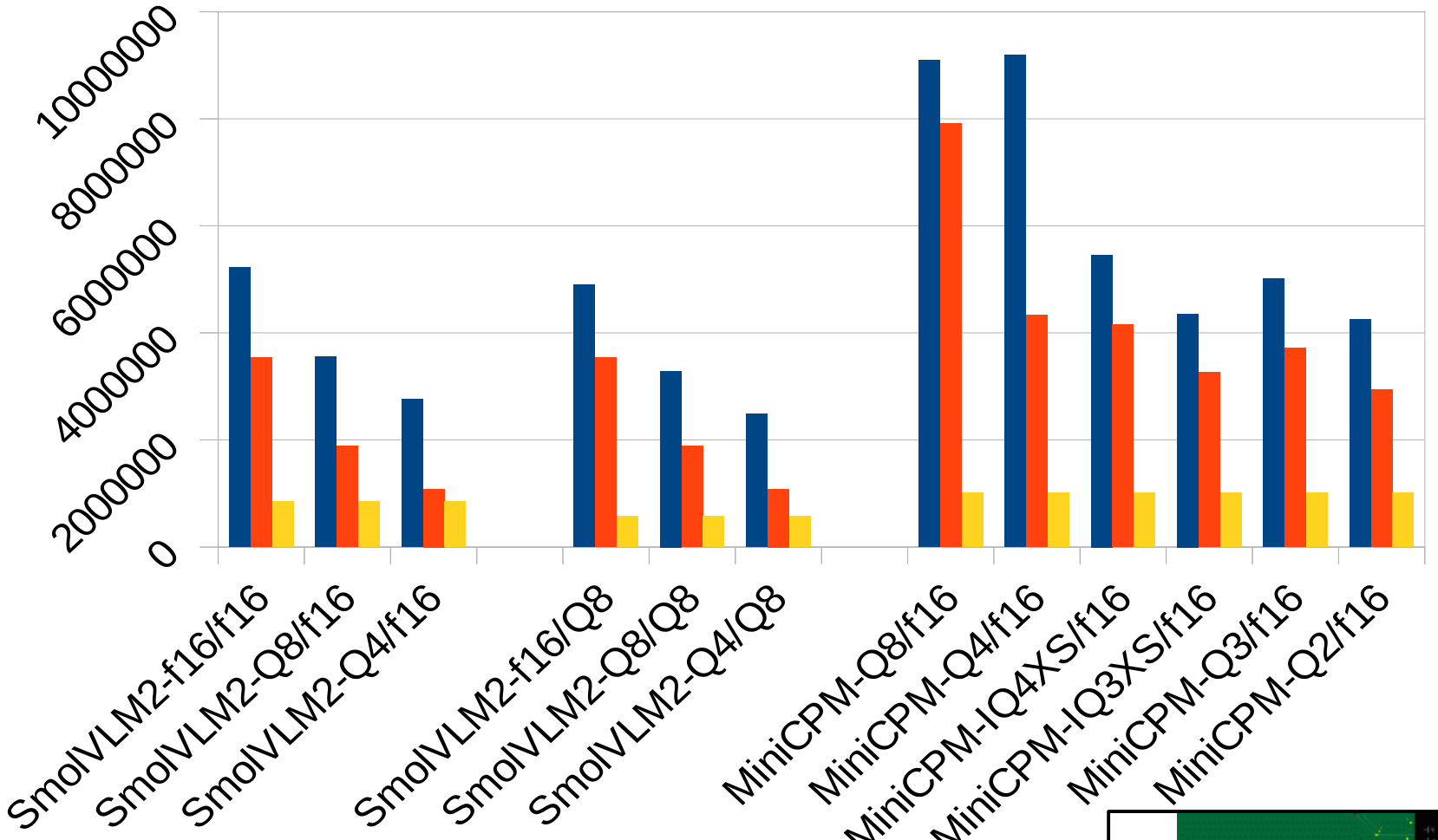
Inference Observations

- Hallucination is a very **real** risk
- More parameter models can do better
- Mmproj quantitation does not have a significant impact

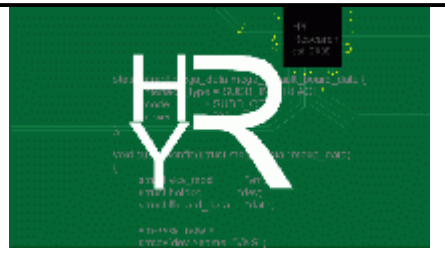


http://www.hy-research.com/

Memory footprint



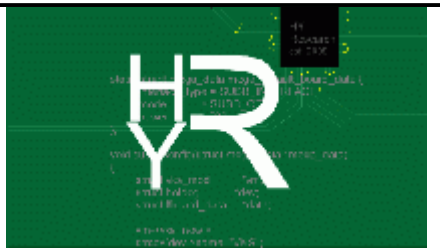
Blue = Measured RSS (from ps)
Red = Main model GGUF file size
Yellow = mmproj file size
Size in Kilobytes



http://www.hy-research.com/

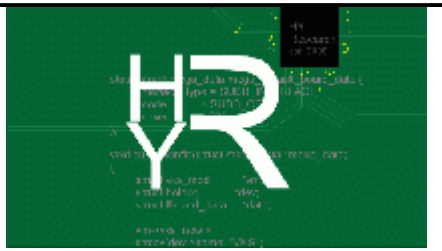
Power

- Limited data
- Both OrangePi5 and RPi5 pulled around 11-12W when inferring about 2W idle.
  - Eyeballed on USB-C power
  - Includes board, LED, networking, and fan



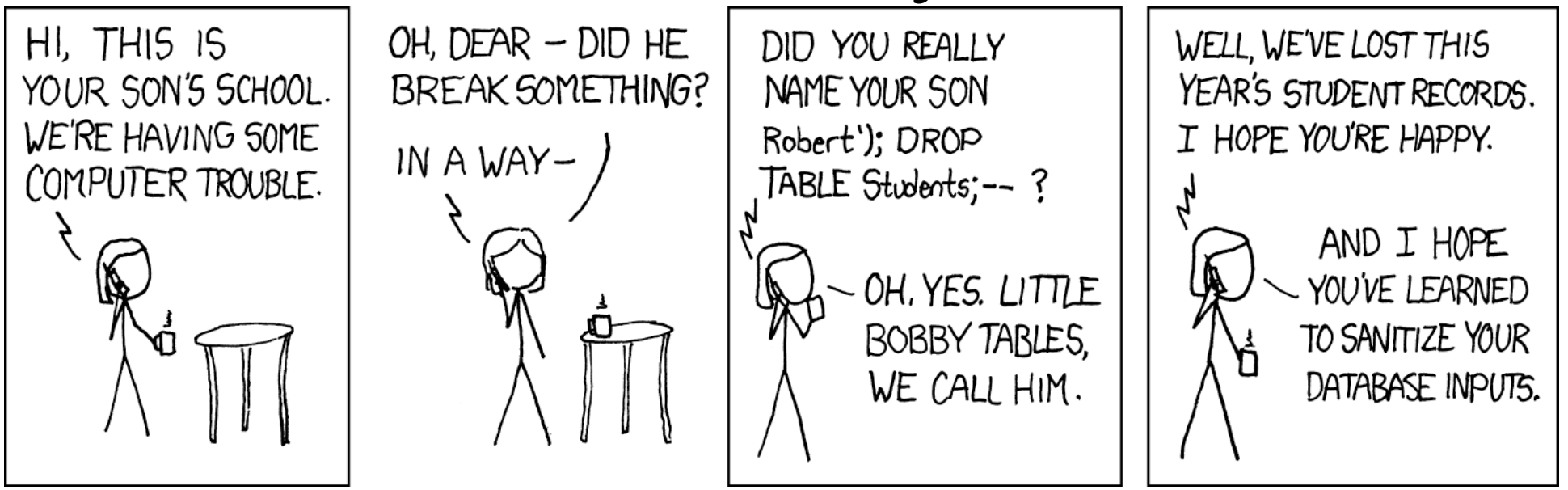
Testing considerations

- LLM inputs are often not a bounded set
  - Clever inputs can trigger unexpected outputs
    - LLM can react to images with non human visible data
- Depending application LLM out can vary slightly from run to run
- Natural language inputs may not be sufficiently covered
  - Perhaps generate test inputs with an LLM?
  - Multi (human) language inputs?
- May have to retest (or even update test cases) if the weights change

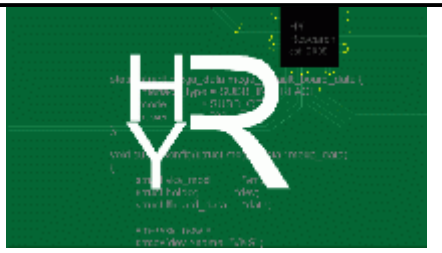


http://www.hy-research.com/

Security



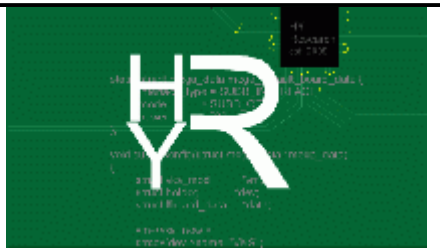
- Many analogs to pre-LLM security
  - Prompt injection
  - DoS
  - Always scrutinize inputs



http://www.hy-research.com/

Security (con't)

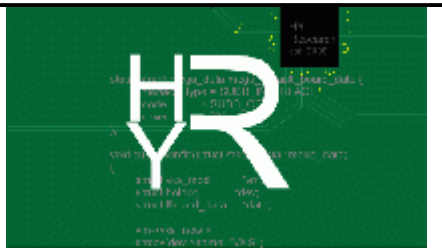
- Be careful with what an LLM is enabled to do (tools)
- Even outputs of an LLM may need to be checked



http://www.hy-research.com/

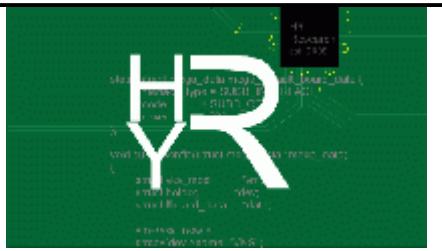
Alternatives

- LLMs are not the only option/method of achieving goals
- Less complex ML models may be applicable
- Other techniques
  - Computer vision algos can look at the 2D spectral output
  - Audio algos can process/analyze spectrum. i.e. Sphinx for speech recognition



Conclusions

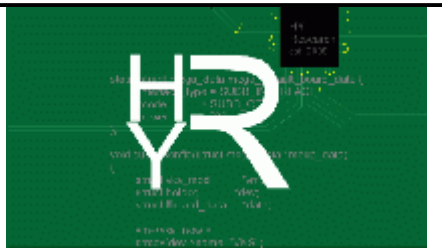
- Embedded Devices can benefit from LLMs
  - Comes at a price (resource, engineering costs. BOM costs)
  - Not suitable for all applications
- Needs special considerations
  - Design in sufficient resources
  - Prepare for functionality from outside of normal code paths (I.e things outside of code scanning)
  - Testing coverage may be less
- Be aware of security
  - Scrutinize all inputs



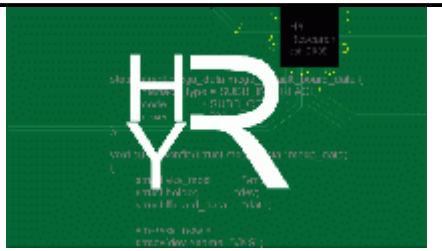
http://www.hy-research.com/

Terminology

- Hallucination – wrong output from an LLM
- Multimodal – trained on media besides text
- Open source – training data, architecture available and open (weights may not be available)
- Open weights – trained weights available and open (model itself may not be open source)
- Tokens – Model specific encoding of input/output
- Tools – An API to enable a LLM to utilize other resources



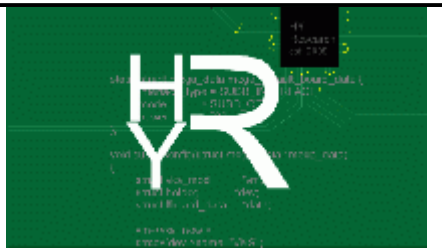
Questions?



http://www.hy-research.com/

Thank you

Slides available on http://www.hy-research.com/



http://www.hy-research.com/