SYNTIANT®

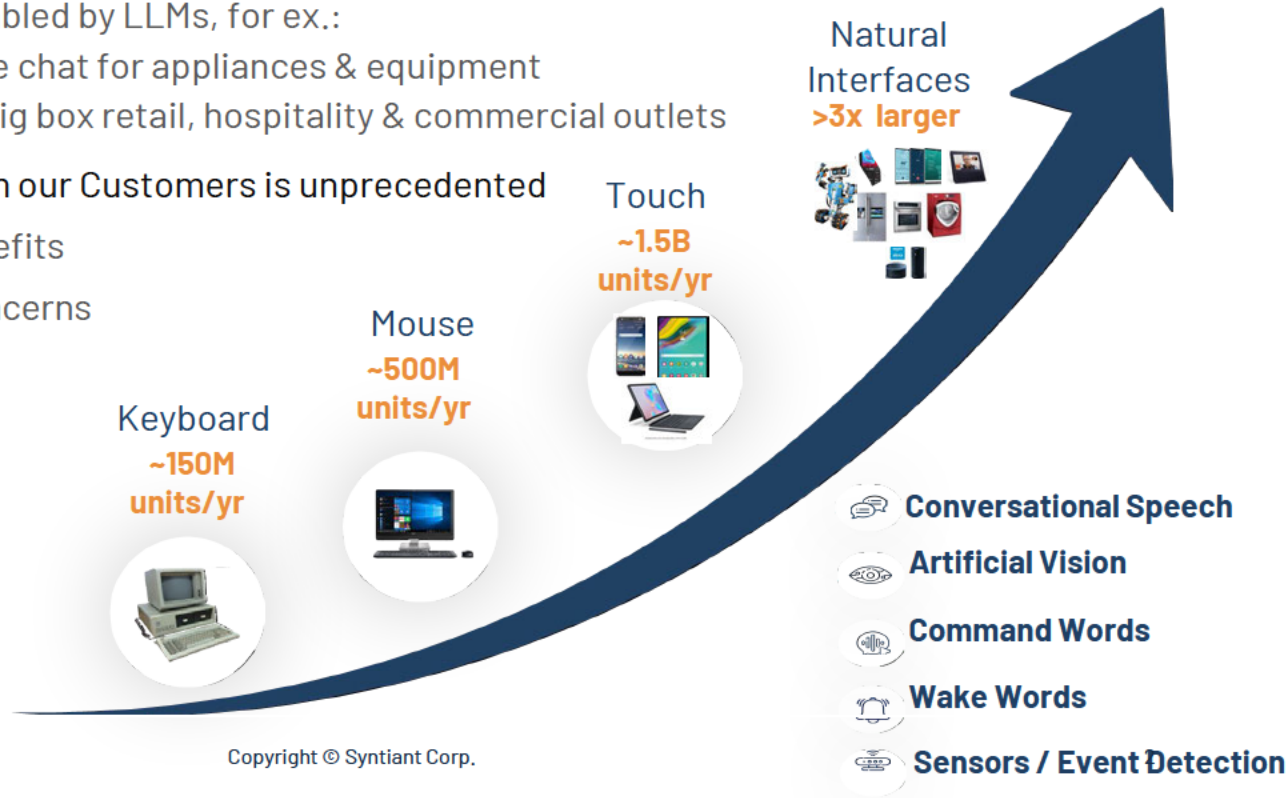
Making Edge AI a Reality

# **Edge of Tomorrow:** Unleashing the Power of Small LLMs for Generative AI at the Edge

Mallik Moturi, Jonathan Su, Stephen Osborne

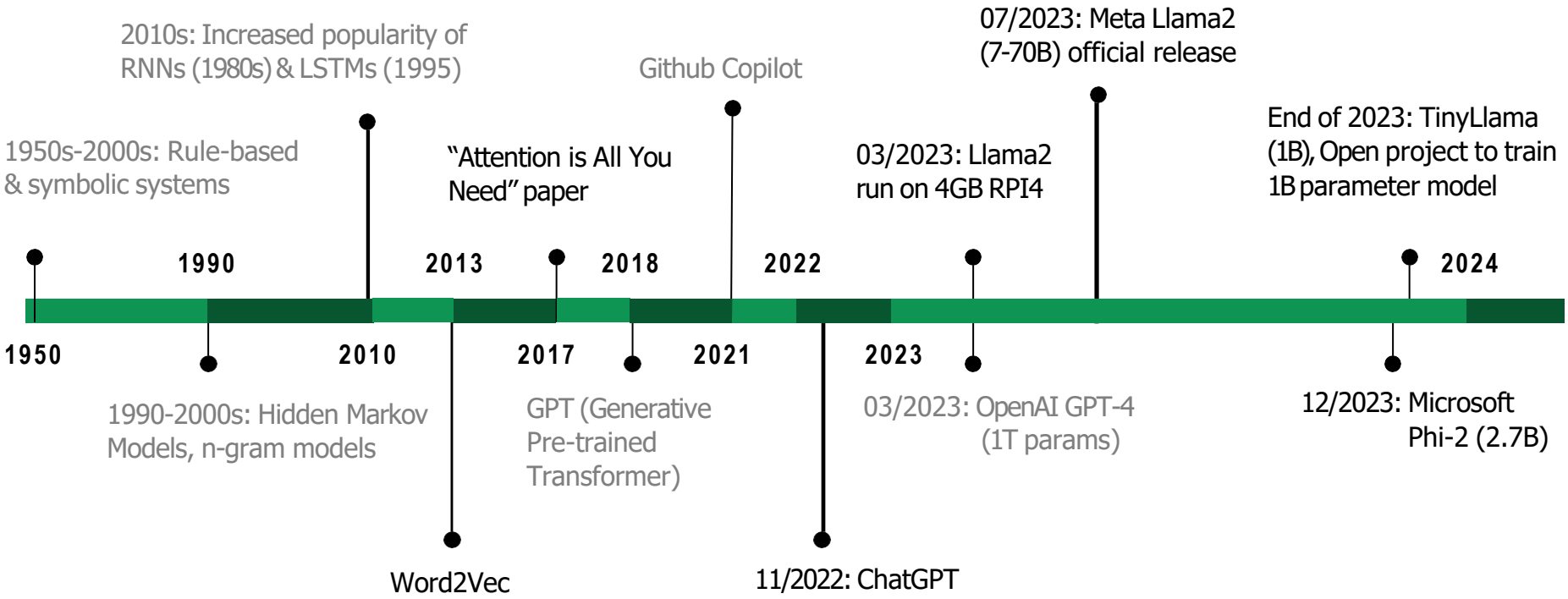
# Why do we run LLMs on edge devices?

- Natural Interfaces enable more applications to bridge the digital world to the real world.
- Conversational Speech, enabled by LLMs, for ex.:
  - Question/answer style chat for appliances & equipment
  - Customer service in big box retail, hospitality & commercial outlets
- Demand for Edge LLMs from our Customers is unprecedented
- Edge devices have many benefits
  - Addresses privacy concerns
  - Improves reliability
  - Lower latency
  - Lower cost
  - Lower energy use
  - More personalization




SYNTIANT

# Large Language Models progress towards the Edge



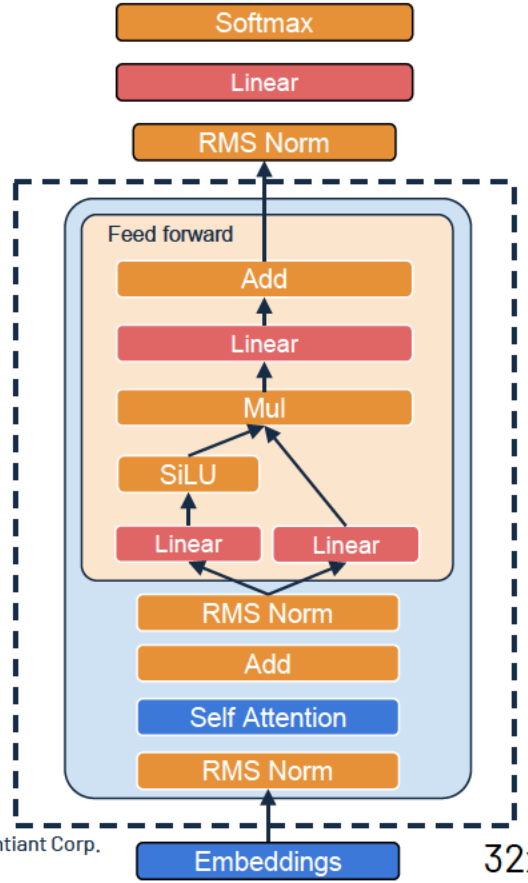
# What is the current state of the art?

- Llama2 and Phi-2 are foundation models with similar architectures.
  - A lot of companies & teams are working on similar models, mainly focussing on the ultra high end
- Seeing a bifurcation with some looking at the low compute regime
  - Qualcomm announced running at 20 tokens / second on a Snapdragon 8 Gen 3
  - Intel announced running at 40 tokens / second on a Xeon Max 9480
  - ARM blog showing 9.6 tokens / second on 3 Cortex-A700 CPUs


But, for real-time LLM applications we require ~ 2.5 words / second
>= 3.3 tokens / second

# Large Language Model Architecture Compute Scaling

- The compute has 2 separate contributions
  - **Linear terms**: Matrix-vector multiplication with the W weight matrices (fixed cost per time step)
  - **Attention terms**: Matrix-vector multiplication with the K key matrix (scales linearly with the number of tokens)
- Example: Llama 2 7B
  - Linear term: 7B MACs per token (1 MAC per parameter)
  - Attention and linear terms are approximately equal cost after ~400 tokens (800 with fused masking)




Feed forward:

$$r = W_3(\text{silu}(W_1 z) \cdot W_2 z)$$

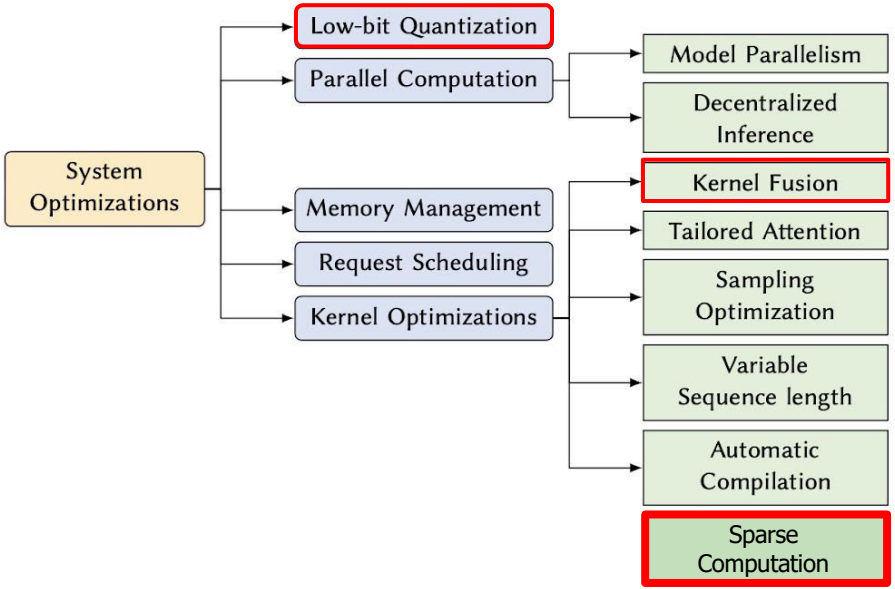
Self Attention:

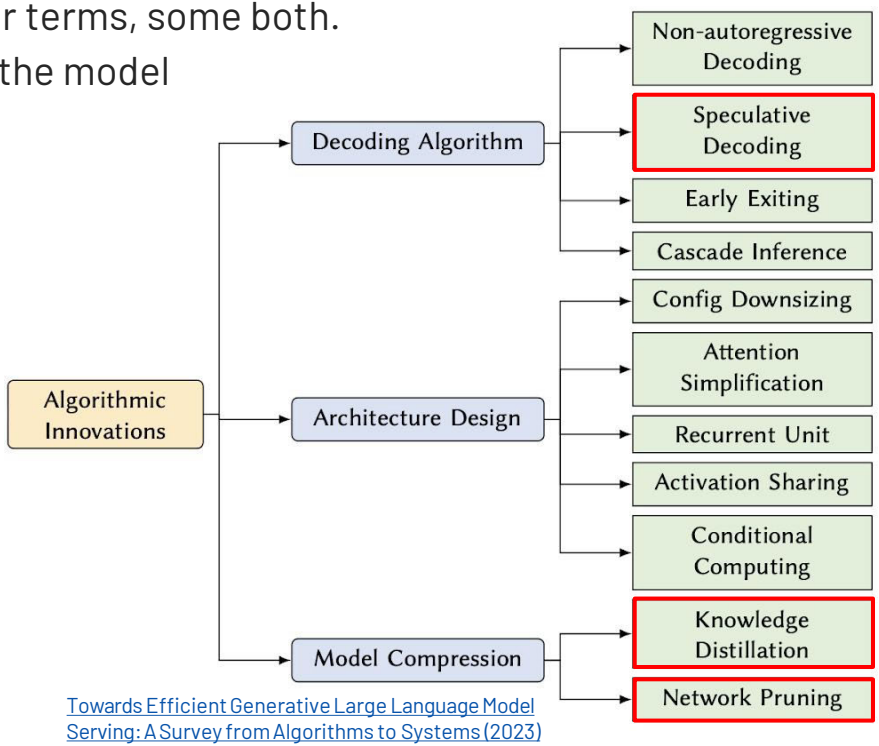
$$q_i = W_q x_i$$
$$k_i = W_k x_i$$
$$v_i = W_v x_i$$
$$y_i = \text{softmax}(q_i K^T / \sqrt{d}) V$$
$$u = Wy$$

SYNTIANT

# Summary of Different Optimization Approaches

- Many approaches, several have been used in other domains for a long time
- Some target the attention terms, some the linear terms, some both.
- Additional benefit from retraining & fine-tuning the model




Towards Efficient Generative Large Language Model Serving: A Survey from Algorithms to Systems (2023)

# Dynamic Sparsification & Retraining

- Switch which parts of the model are evaluated depending upon the input
- If model uses ReLUs then naturally get sparsity. If not, can still approximate small values as 0
- ~1.3x speedup without much loss in accuracy
- Many more optimizations to reduce amount of data
  - input sparsity, output sparsity, efficient data structures for memory management, row-column binding

- For edge applications we did supervised fine-tuning (SFT) on a custom dataset with modest resources (16 A100 GPUs)
- This allows additional optimizations, e.g.
  - Quantization-aware training
  - Sparsity-aware training
- Leading to 2x speedup

# LLM Sparsification Speedup example

Comparison of the previous state-of-the-art GGML LLaMa-7B implementation (right)

Vs.

Syntiant optimized LLM (left), with the sparsified version outputting tokens at **2x the speed** with the **same accuracy**.

**Syntiant**
**2x Faster**

**Previous SotA**
**(llama.cpp)**

# Summary: Syntiant is Accelerating LLMs for the Edge

- Like the rest of the AI industry, demand for LLMs from our customers is huge
- Unlike the rest of the AI industry, we operate in compute constrained environments
  - Leveraging our expertise in sparse edge computation, we have developed a generalized sparsity approach that speeds up the State of the Art LLaMA-7B model by 1.3x – 2x
  - On a multi-threaded x86 machine, this means a rate of 10 tokens per second.
  - On edge-accelerated NPU's supported by our Syntiant Inference SDK (Ambarella, Qualcomm, etc), we achieve up-to 30 tokens per second.
- These are the customer use-cases we are enabling:
  - Question/Answer style chat for home appliances, commercial equipment, etc.
  - Customer service in big box retail
- It is our belief that optimized LLMs running on Edge hardware will gain widespread adoption.

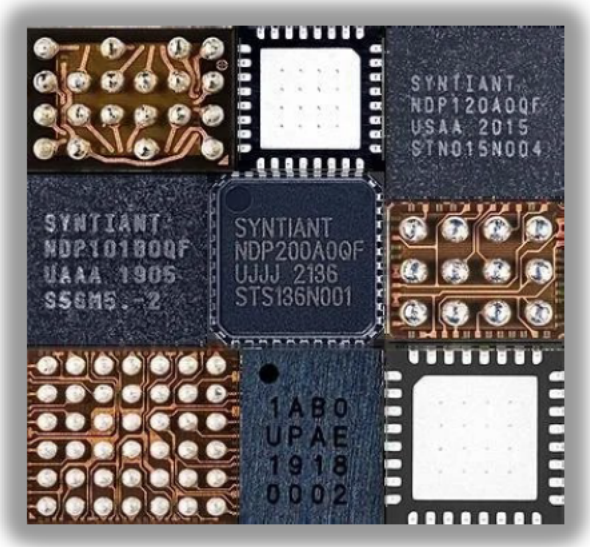
# Path Forward: Shaping the Edge of Tomorrow with Small LLMs

**Edge-Optimized LLMs**: Tailoring small LLMs for more intuitive, natural interfaces for real-world to AI interactions

**Next-Gen Silicon**: Custom silicon compute & memory will boost efficiency and generative AI capabilities at the Edge

**Economizing Memory**: Innovation in memory usage, shrink both physical & cost footprint of Edge LLMs & generative AI

**Generative AI Ubiquity**: Forecasting a future where small, efficient LLMs are ubiquitous, transforming everyday technology with natural, generative interfaces.



# Thank You


info@syntiant.com

@Syntiantcorp

www.syntiant.com

Syntiant Corp.


SYNTIANT