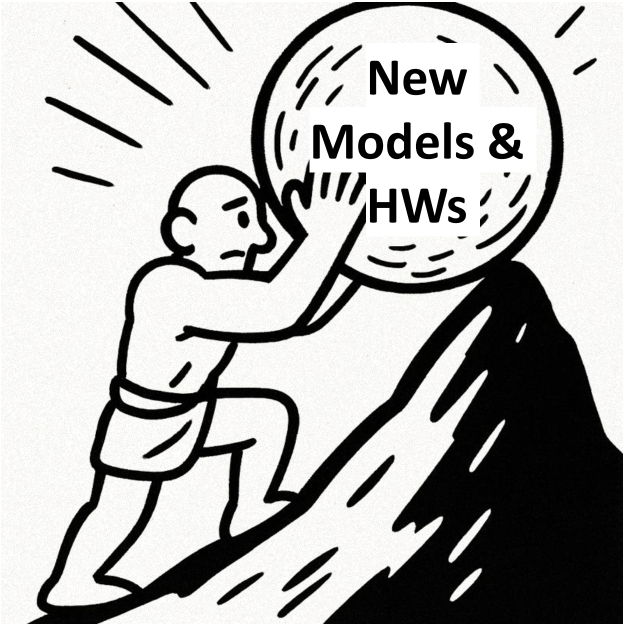
Bridging the Gap: Streamlining the Process of Deploying AI onto Processors

Taesu Kim

CTO

SqueezeBits Inc.

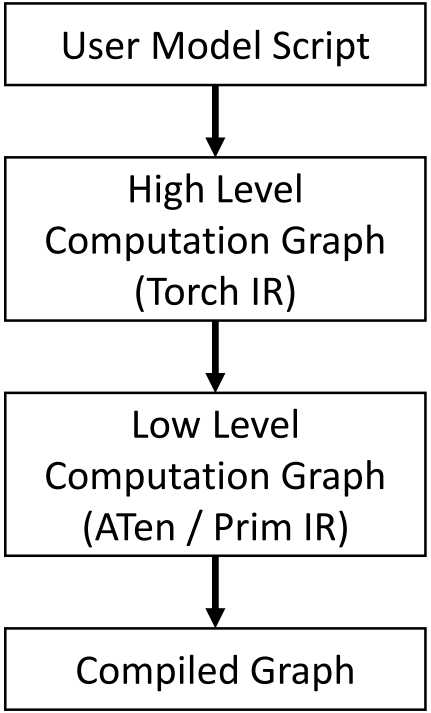
The Challenge of AI Deployment



- Supporting diverse models
  - Computer vision
  - Larger models (LLMs, diffusion …)
- Multiple hardware targets (GPUs, Mobile, ..)
- Manual conversion scripts needed

- **Innovation is getting slowed down**

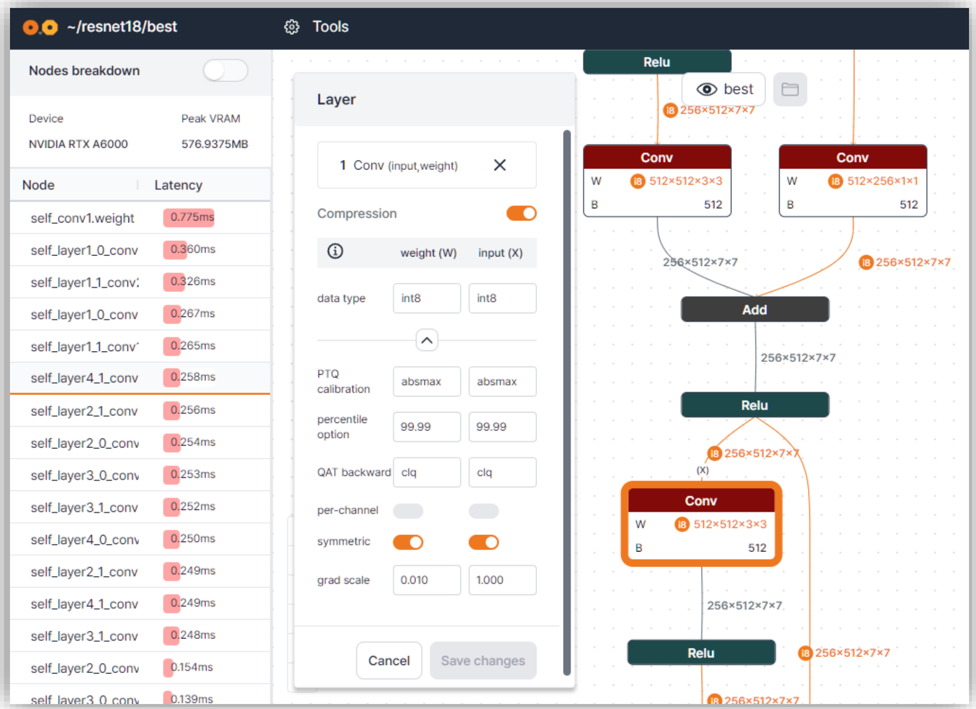
Model-Agnostic Conversion Process



- PyTorch 2.0 with several tools to support model-agnostic deployment
  - TorchDynamo: Python-level just-in-time compiler
  - TorchInductor: Fast codegen with loop level IR
  - AOTAutograd: Ahead-of-time graph tracer / deep learning compiler integration

- **Robust and fast, but sometimes harder to use**

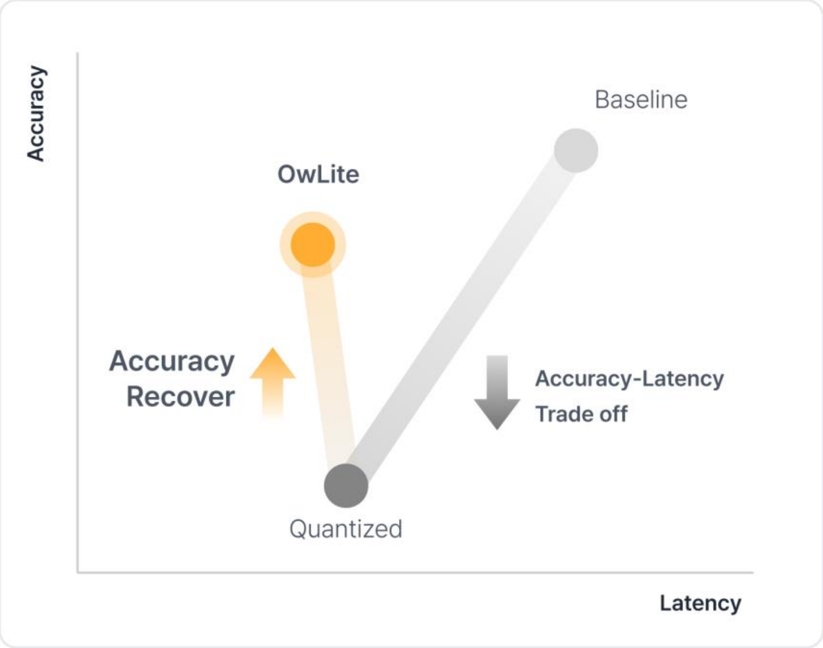
Our solution: OwLite



- Native integration with PyTorch
- Supports all PyTorch operators
- Multiple precisions, formats, and quantization algorithms
  - E.g., INT8, FP8 (E4M3, E3M4)
- Layer-wise fine-grained quantization
  - Applicable through simple UI

SqueezeBits

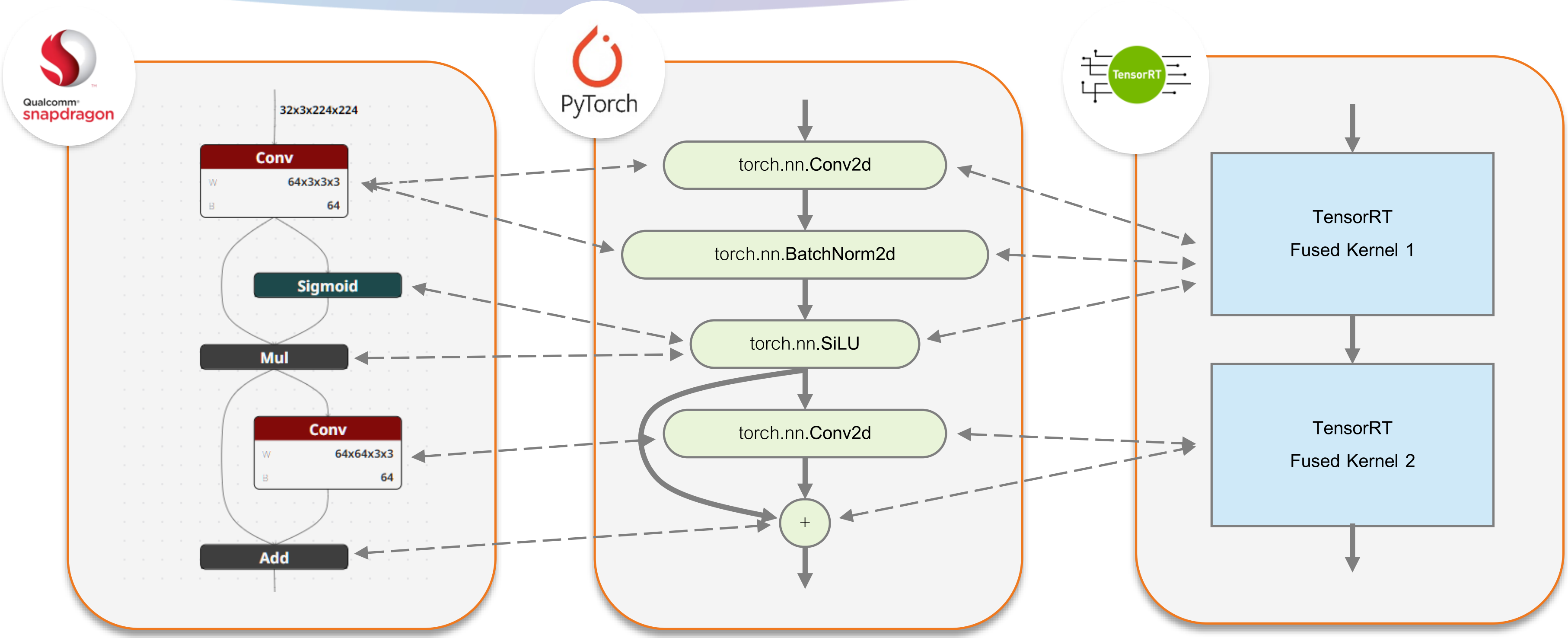
Our solution: OwLite



- **Quantization-aware-training support**

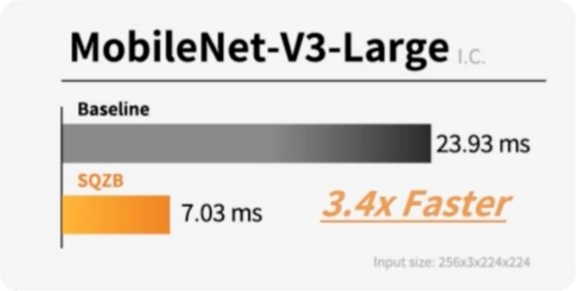
- Compressed models can be trained again for accuracy recovery!
- Users can reuse their own data loader and training scripts.
- Fine-tuned models can be deployed to target devices with same configuration.

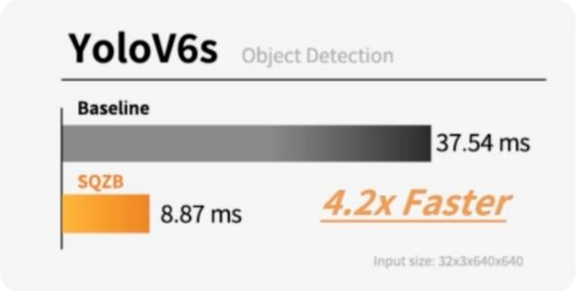
Our solution: OwLite



Supports Diverse Hardware

OwLite in Vision Applications

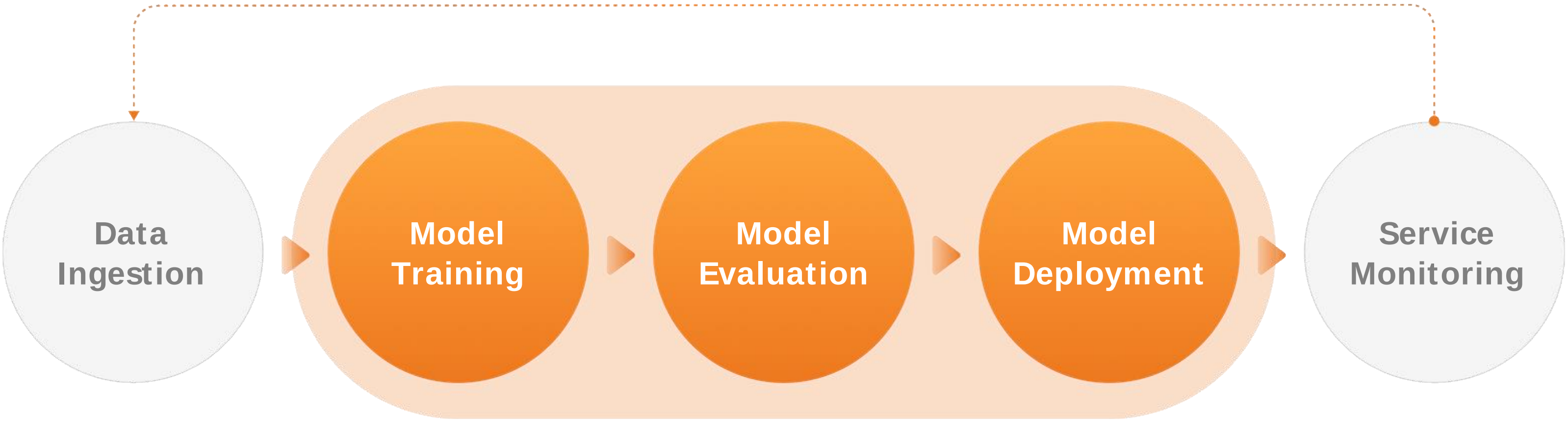




(Tested on a NVIDIA A6000, TensorRT)

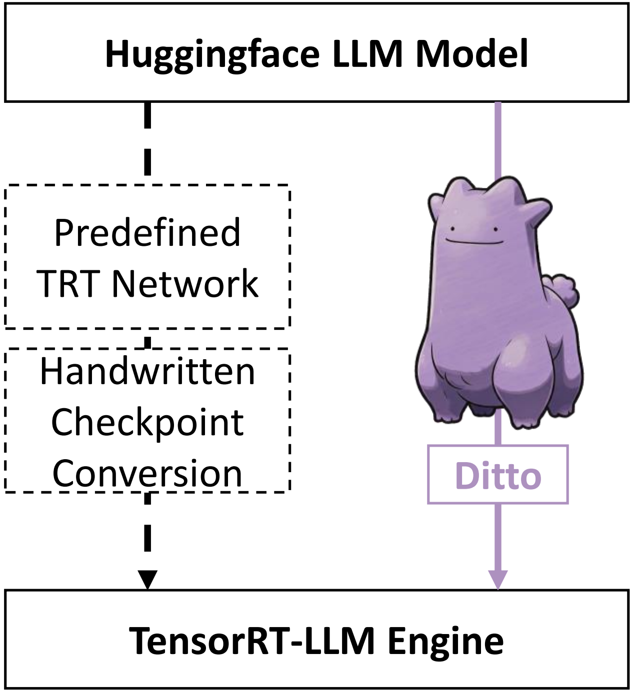
- Available tasks (examples):
  - Image classification, object detection, image segmentation, text classification, re-identification, face landmark, pose estimation, and many more
- Supports up to 1B parameter models
  - Models with too many nodes to visualize are currently not supported.
- Bring your own model!
  - Even supports transformer-based ones!

Consider Deployment from Model Training Stage



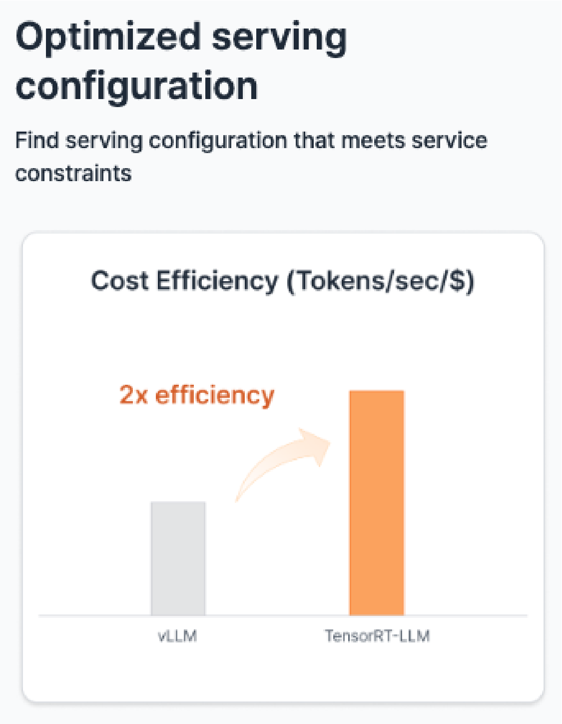
- Models must be trained considering their performance upon deployment.
  - Larger models with low precision can outperform smaller models.
- Rapid prototyping and validation are crucial.

Ditto: Model-Agnostic Converter for LLMs



- Model-agnostic converter for LLMs
  - Currently supports TensorRT-LLM for NVIDIA GPUs
  - No need for hand-coded conversion script!
- Converts models in *Transformers* library to TensorRT-LLM engines
- Diverse graph optimizations to support LLM-specific features

Fits on Chips: Revolutionizing LLMs Deployment



- **“Click, Benchmark, Deploy.”**
- Diverse serving frameworks & hardware
  - vLLM (NVIDIA GPUs, Intel Gaudi)
  - TensorRT-LLM (NVIDIA GPUs **with Ditto**)
  - More to come (sglang for GPUs, etc.)
- Tool for non-expert users
- **Helps optimize LLM serving – reduce your LLM serving cost!**

Conclusions

- **Reduce development time** with model-agnostic deployment pipelines.
- **Optimize performance** by embedding deployment considerations into the training stage.
- **Cut serving costs dramatically** by exploring a wide range of configuration options.
- **Leverage existing tools** to streamline and accelerate your deployment workflow.

Try It Now!

EMBEDDED VISION SUMMIT

OwLite
light, still all right

Fits on Chips

- Our deployment pipelines are being served as both open-source software and SaaS toolkits.
- Start deploying your own models today with **OwLite** and **Fits on Chips**
  - OwLite has free-tier offers for developers (come visit us at our booth #817!)
  - Fits on Chips is being served as free. Try it now!

SqueezeBits

Resources

OwLite (Quantization and Deployment) https://owlite.ai

Fits on Chips (LLM Deployment) https://fitsonchips.ai

Torch-TRTLLM (Ditto, Open Source) https://github.com/SqueezeBits/Torch-TRTLLM

SqueezeBits Tech Blog https://blog.squeezebits.com

Come visit us at booth #817 for demo!