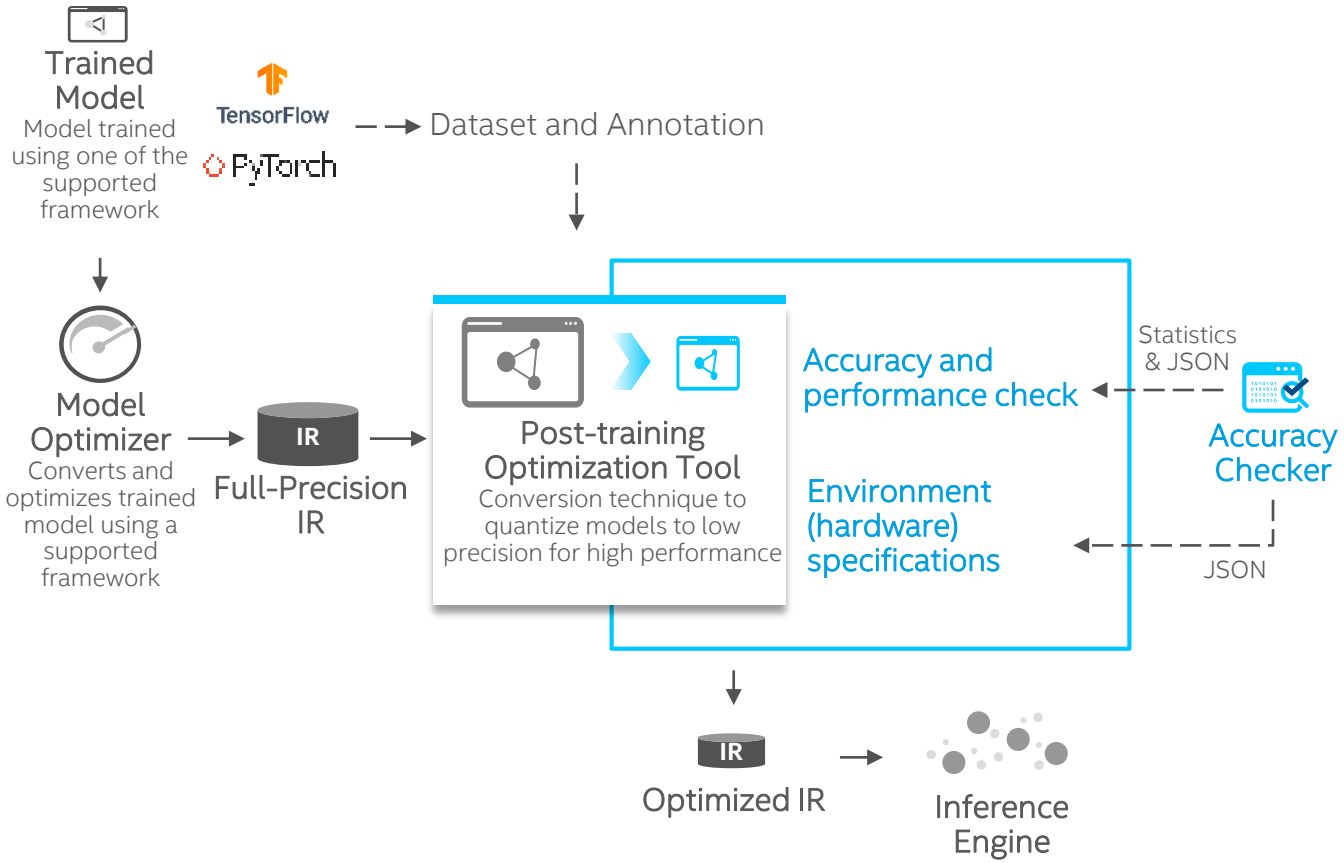
# Post-Training Optimization Tool

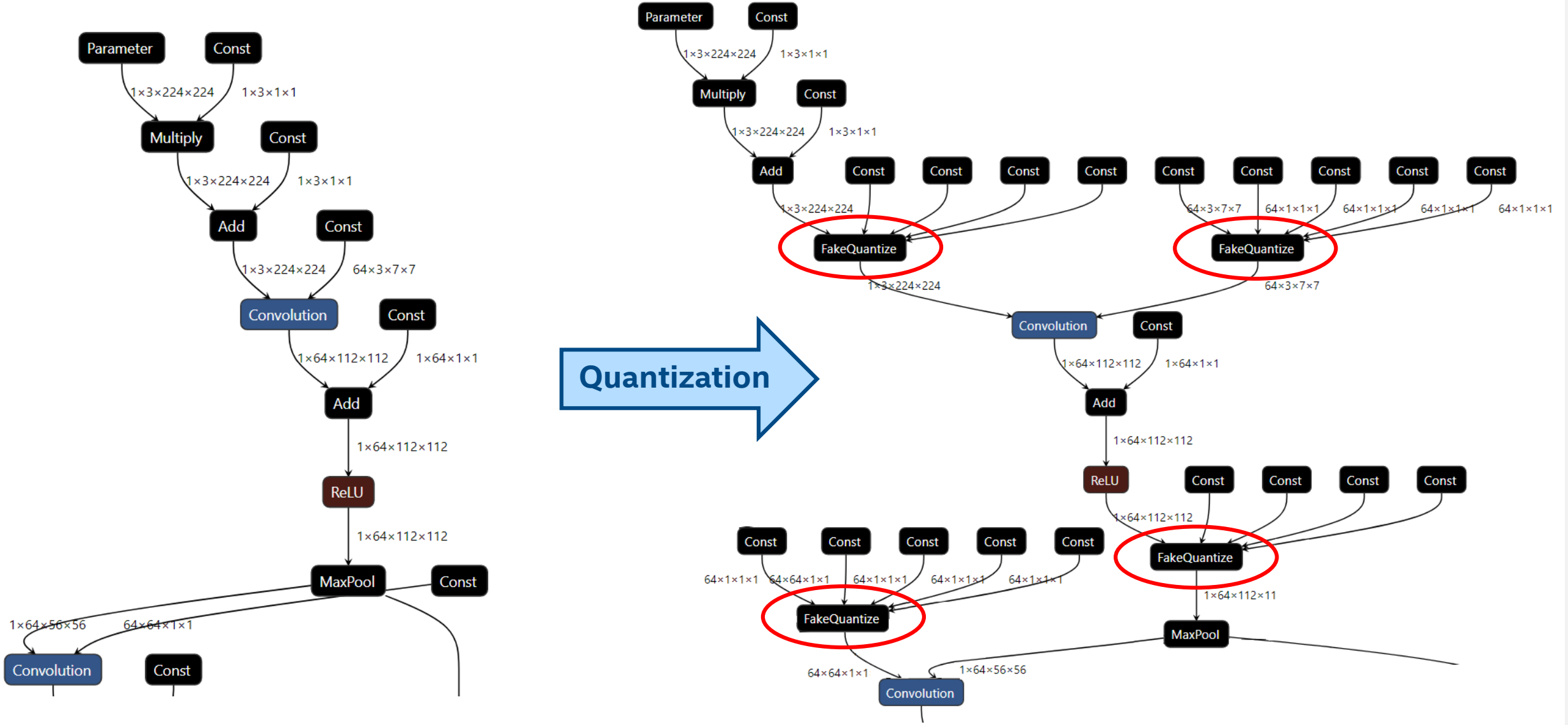
# Post-Training Optimization Tool

- Using the Python API, the Post-training Optimization Tool integrates with the Model Optimizer, DL Workbench and accuracy checker tools to streamline the development process
- Enables a conversion technique of deep learning model that **reduces model size into low precision data types**, such as INT8, without re-training
- Reduces model size **while also improving latency, with little degradation** in model accuracy and without model re-training.
- Different optimization approaches are supported: quantization algorithms, sparsity, etc.

**Performance Benchmarks**
https://docs.openvinotoolkit.org/latest/_docs_performance_int8_vs_fp32.html



# IR transformation

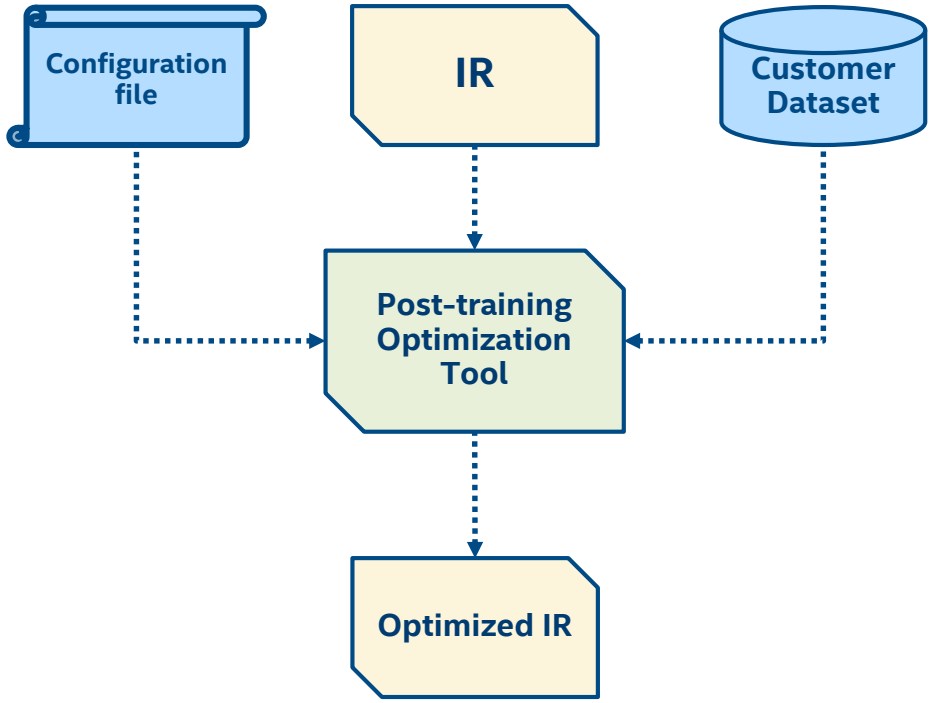


# Post-Training Optimization Tool – features

- Supports quantization of OpenVINO™ toolkit's IR models for various types of Intel® hardware
- *Learn more:* https://docs.openvinotoolkit.org/latest/_compression_algorithms_quantization_README.html
  - Two main algorithms supported and exposed through Deep Learning Workbench:
    - Default algorithm: essentially a pipeline running three base algorithms:
      i. Activation Channel Alignment (applied to align activation ranges)
      ii. MinMax
      iii. Bias Correction (runs atop naive algorithm; based on minimization of per-channel quantization error)
    - Accuracy-Aware algorithm: preserves accuracy of the resulting model, keeping accuracy drop below threshold
  - Provides hardware-specific configurations
  - Features per-channel/per-tensor quantization granularity
  - Supports symmetric/asymmetric quantization through presets mechanism

# Usage scenarios

1 Used as-is. Command line/Workbench scenarios.




2 Integration in user pipeline.