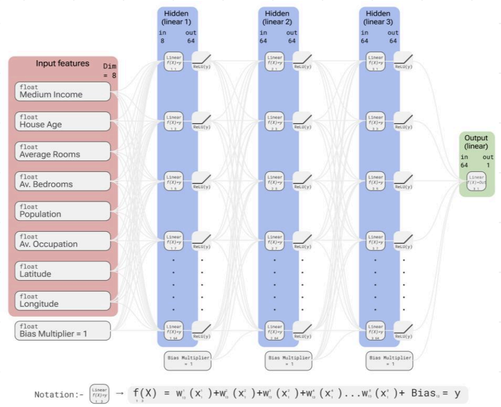
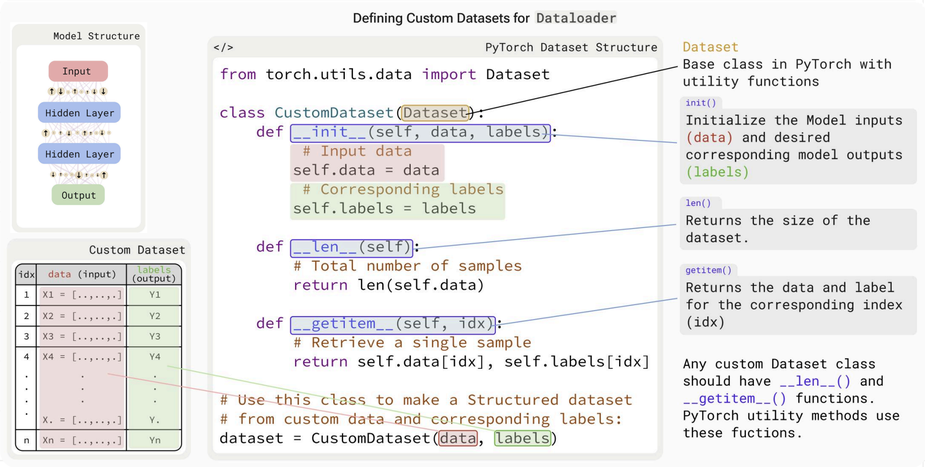
PyTorch
IBM Foundation Model Stack

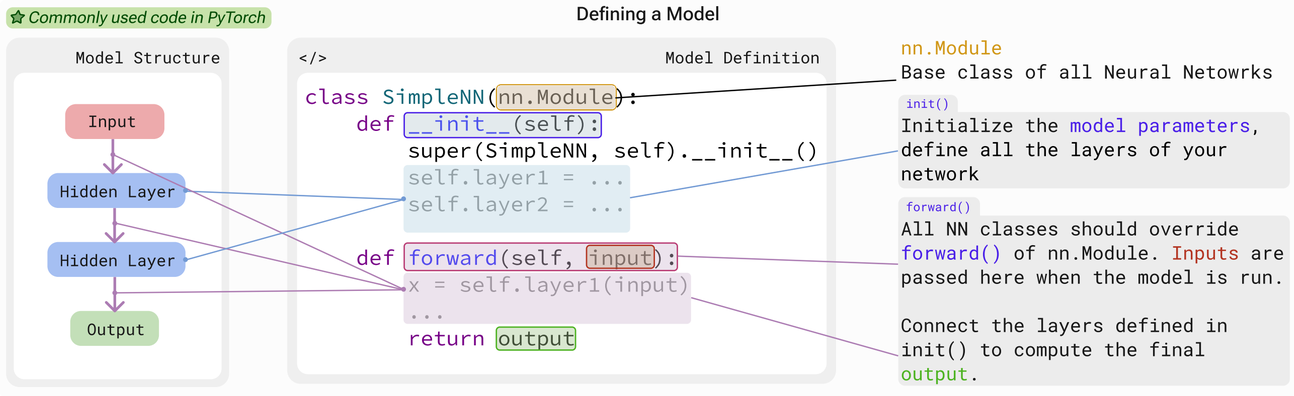
Niraj Kamal Karunanidhi
MS in Computer Science
IBM Open Source AI Intern

Georgia Institute of Technology

Pytorch







- **PyTorch labs for PyTorch Foundation's Certification**
  - Building Neural Networks with PyTorch
  - Benchmarking Models
  - Leveraging Automatic Mixed Precision for training and inference
  - Activation Functions for Models
  - Performance Profiler for Models
  - Creating Neural Network Checkpoints

- ⭐ **Recognized as a Top contributor to the PyTorch Docathon 2025**
- **New changes** to APIs in Automatic Mixed Precision and ONNX unified model format. Contributions to migration to new Sphynx documentation.
  - 6 Pull Requests Created (4 Merged) **800+ lines each**
  - Reviewed 8 Pull Requests

IBM Foundation Model Stack

Modules

- Attention
- Feed Forward NN
- Classification head
- Linear Layers
- Positional Encoding
- State Space Model
- Tensor Parallel

Model Architectures

- Granite
- Llama
- Mistral

.
.
.
.

- Mamba

Optimized Implementations

Training Optimizatons

CPU only,
GPU
Multi GPU

Inference Optimizations

e.g.,
Use KV Cache

Training

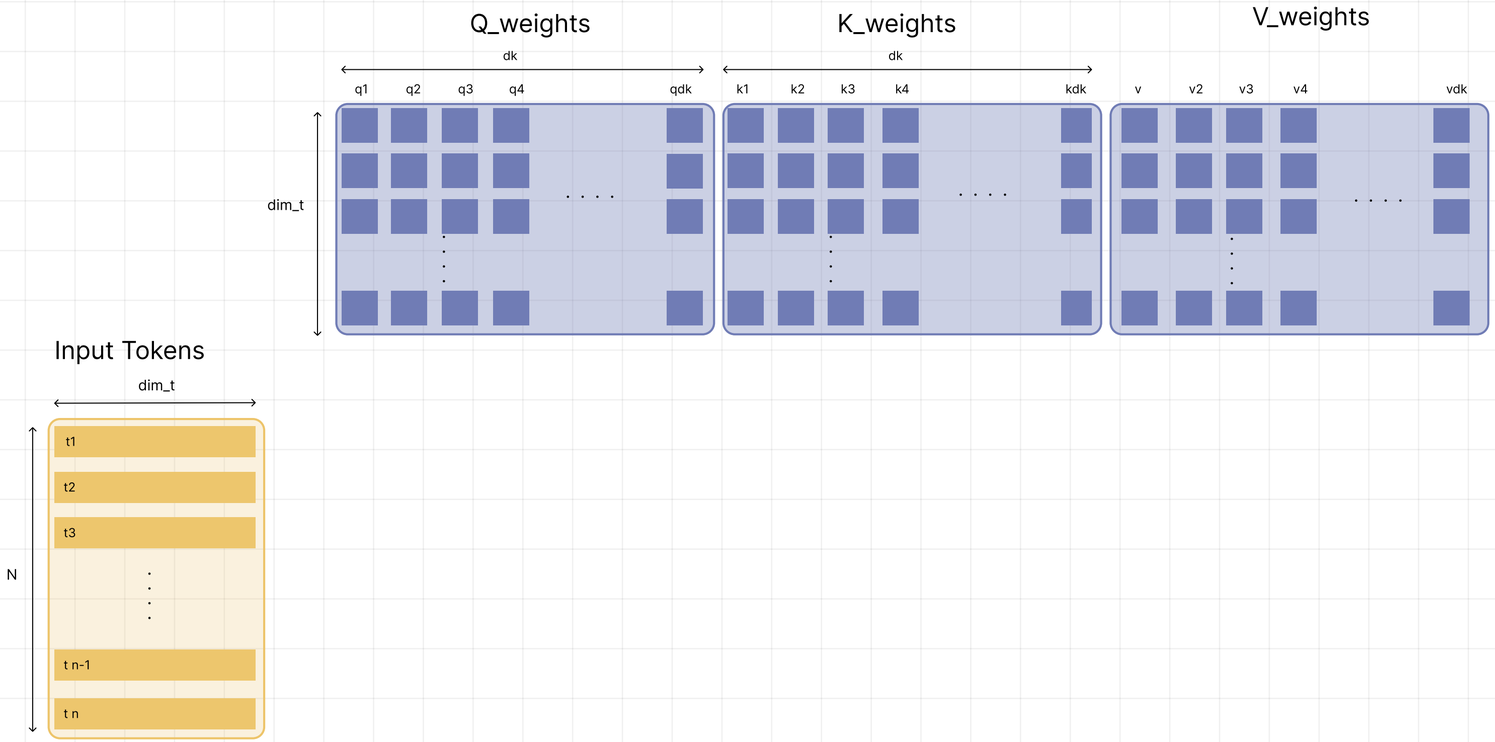
Freezing Layers, changing, curriculum, hyperparameters

Testing

- Performance
- Correctness of Optimizations

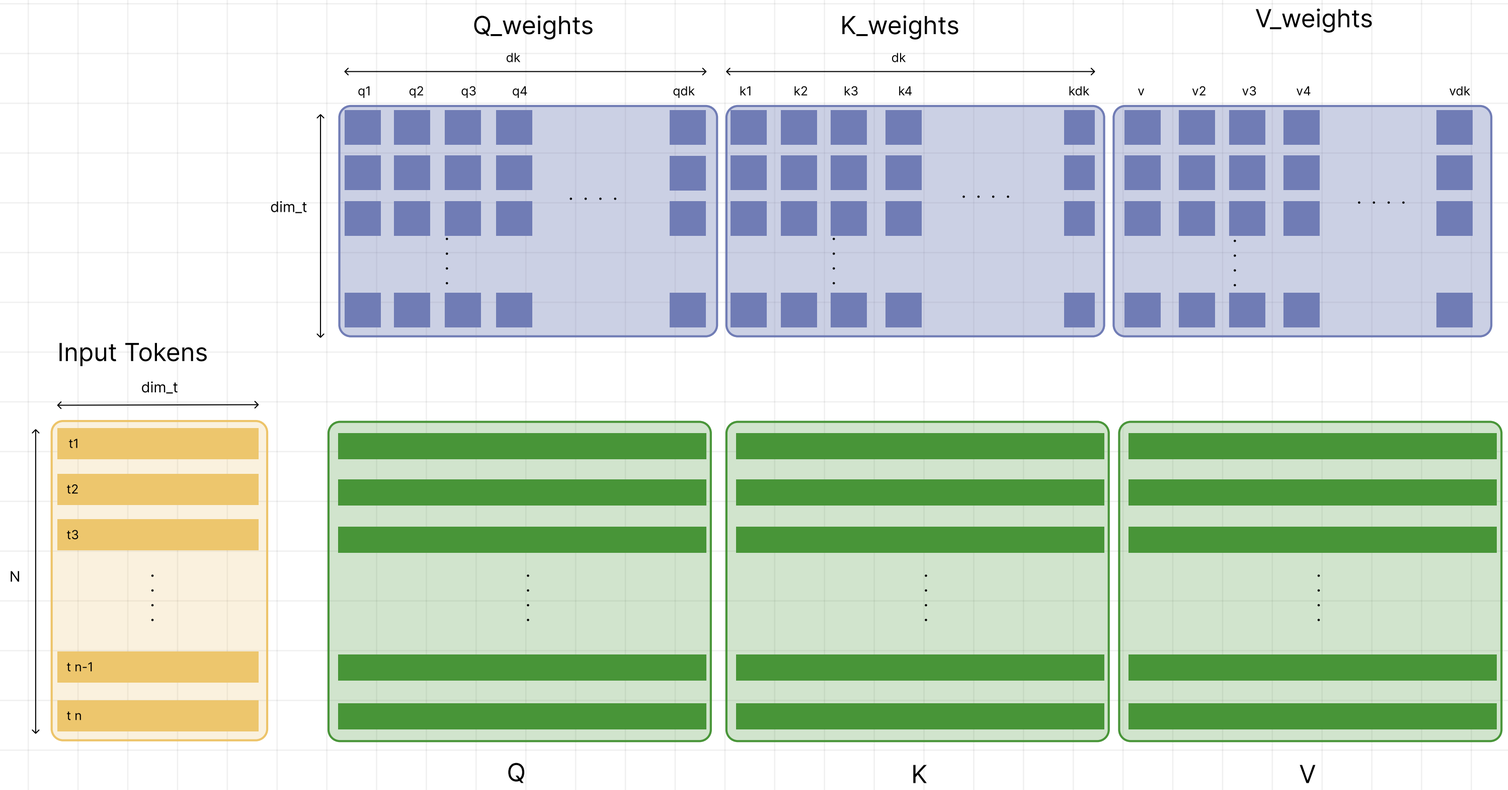
IBM Foundation Model Stack

Example: Attention (Normal Implementation)



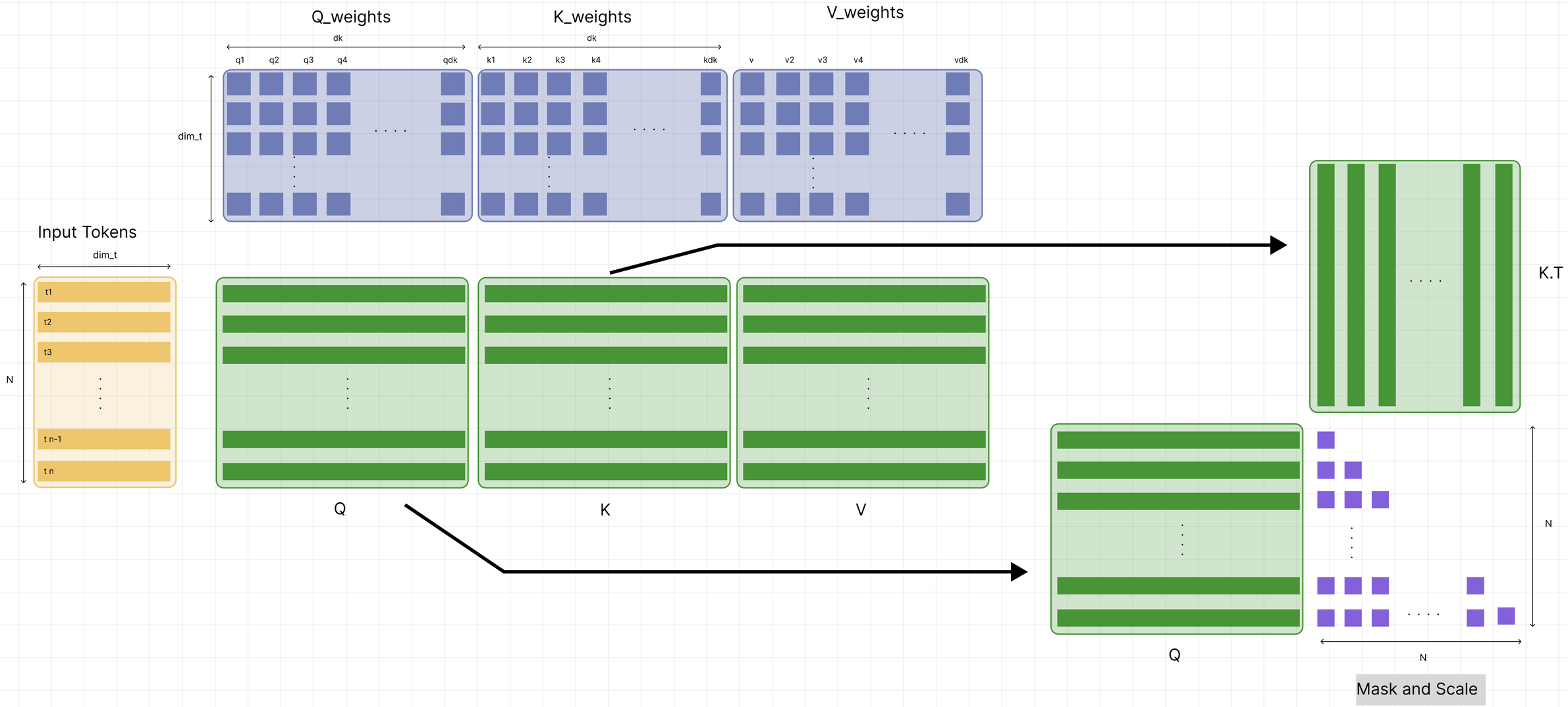
IBM Foundation Model Stack

Example: Attention (Normal Implementation)



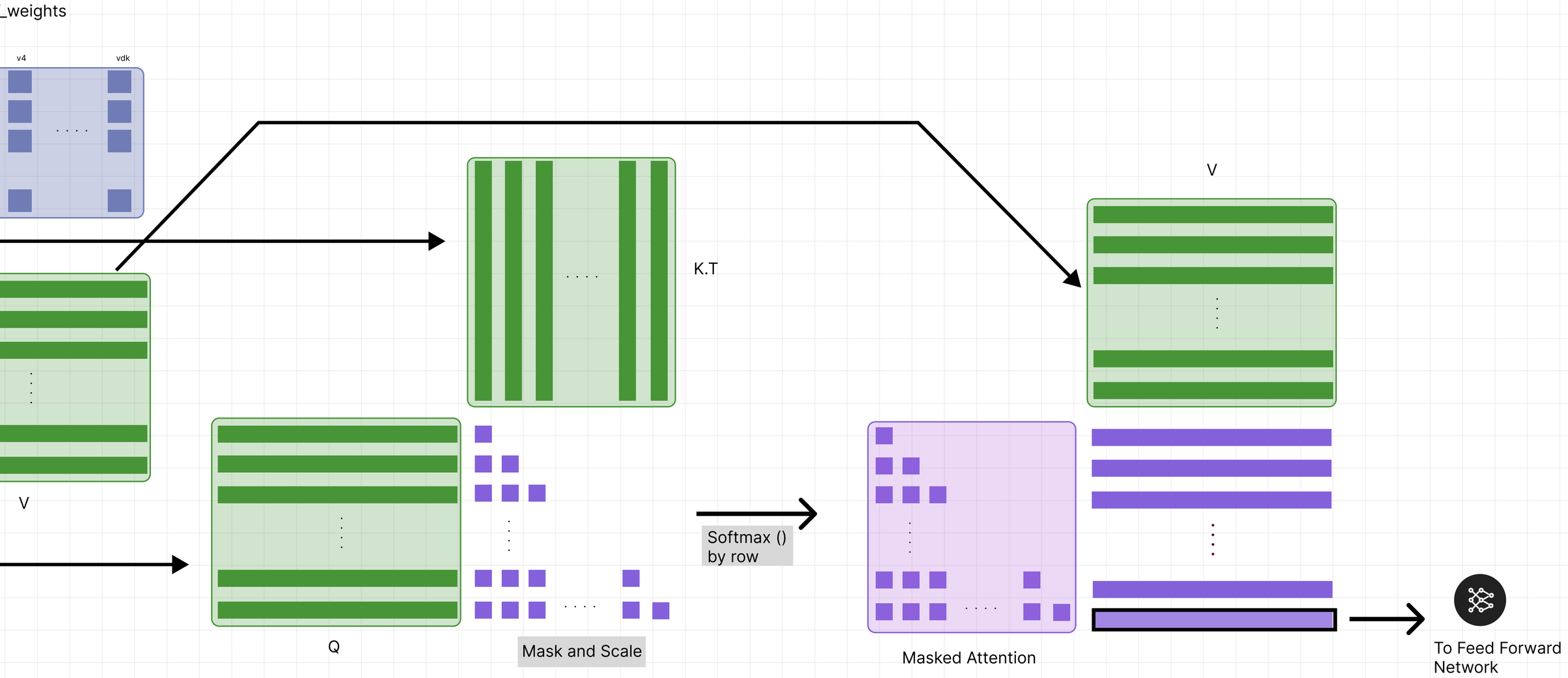
IBM Foundation Model Stack

Example: Attention (Normal Implementation)



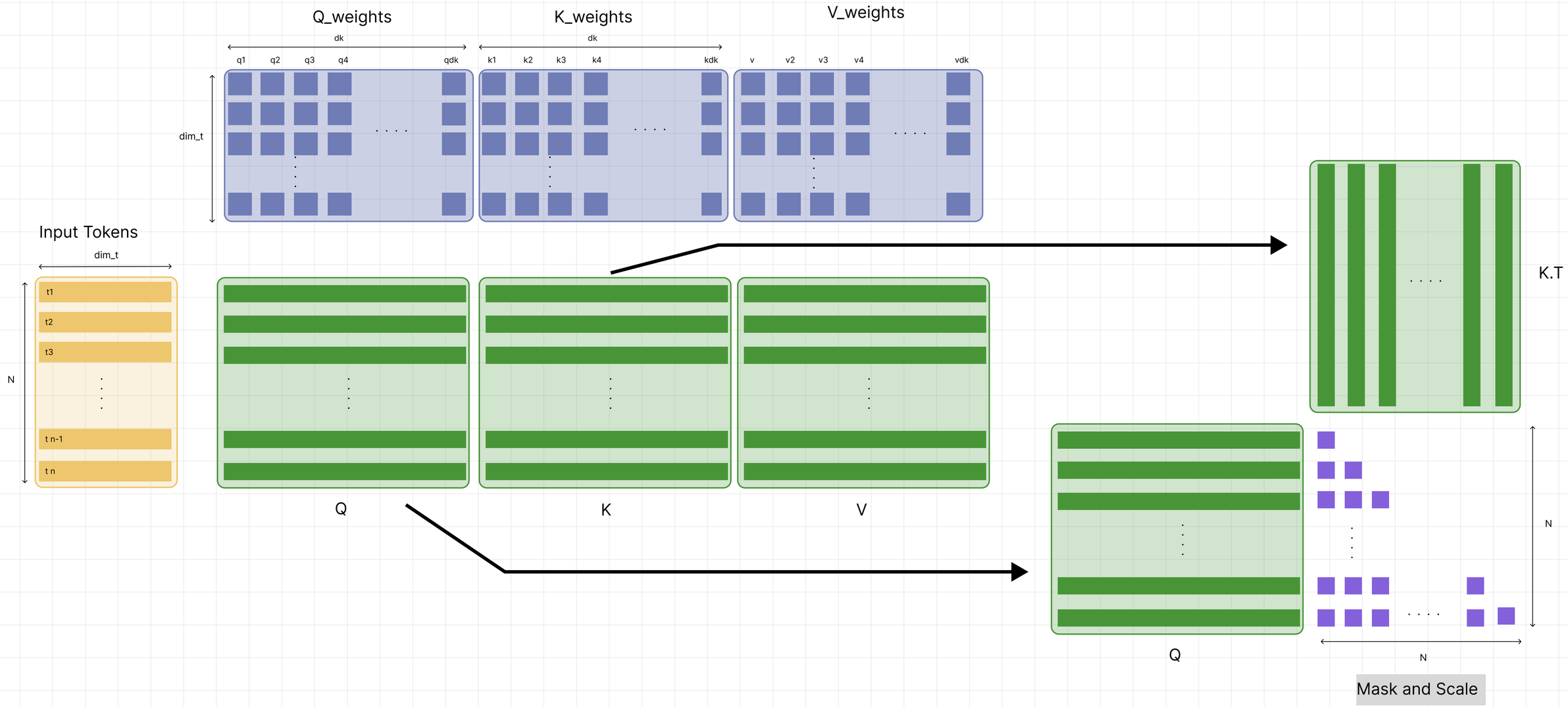
IBM Foundation Model Stack

Example: Attention (Normal Implementation)



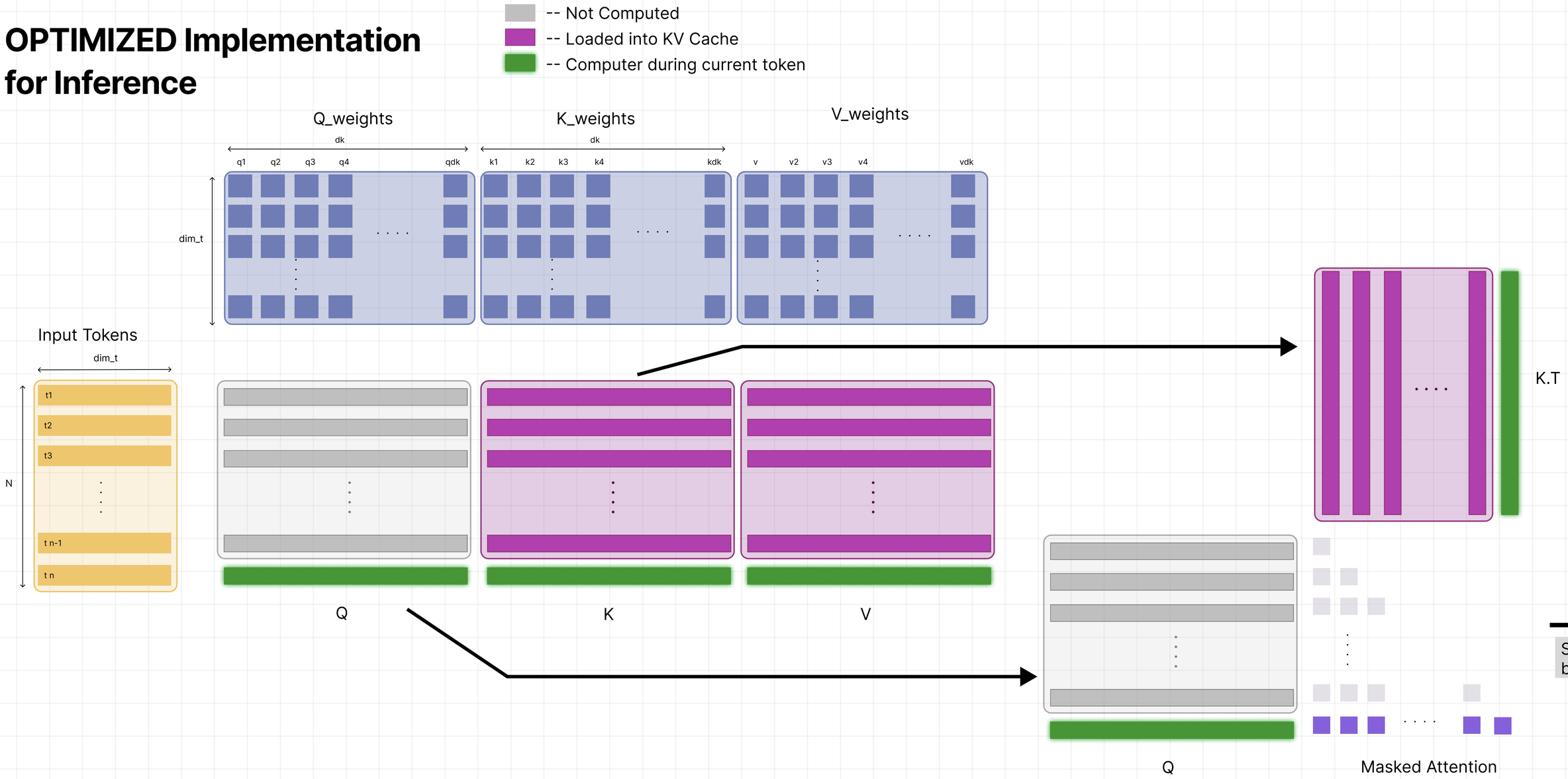
IBM Foundation Model Stack

Example: Attention (Normal Implementation)



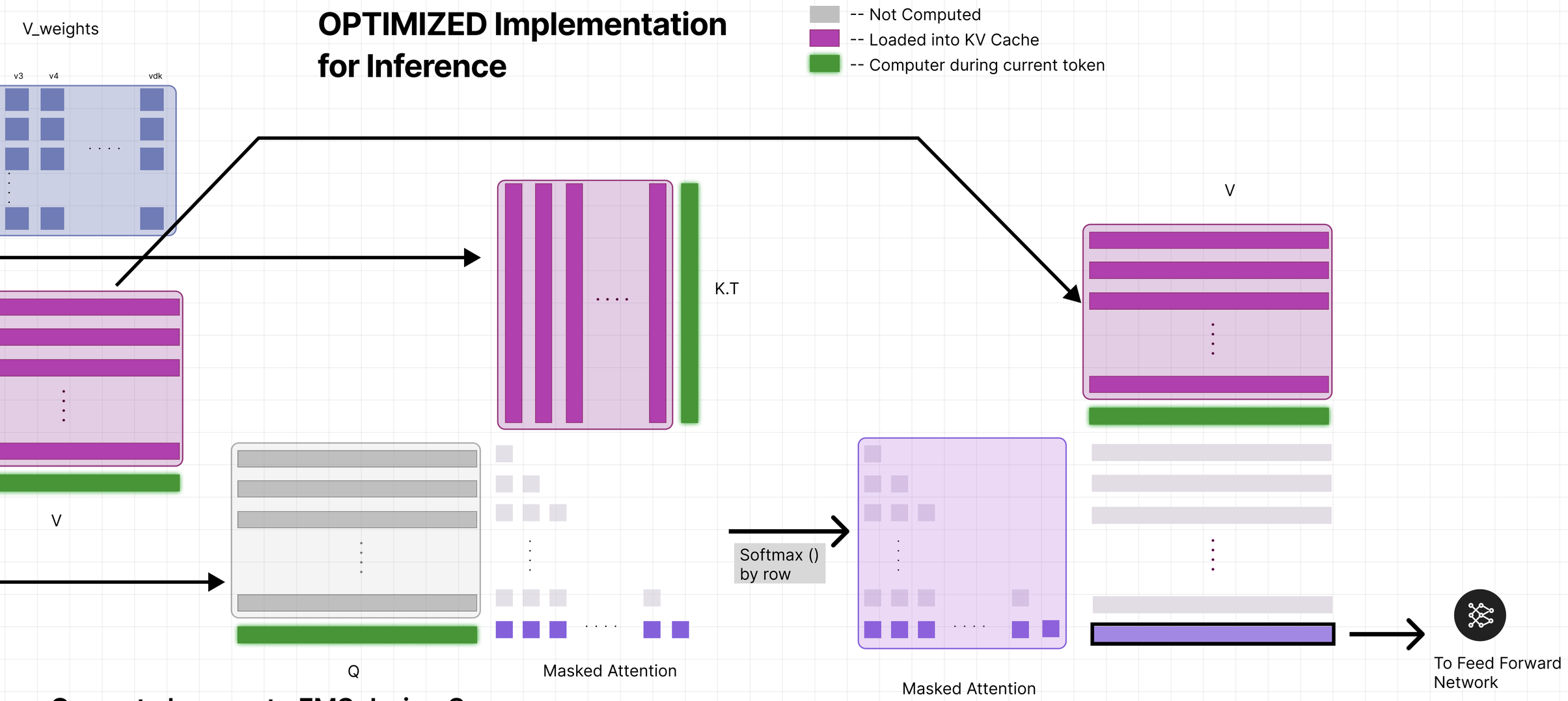
IBM Foundation Model Stack

OPTIMIZED Implementation for Inference



IBM Foundation Model Stack

OPTIMIZED Implementation for Inference



Current changes to FMS during Summer:

- **Added support for Decoder based LLM Model verification - for Prefill and Decode Stages.**
- Improved the handling of **generative models** (e.g., causal language models like LLaMA) while maintaining backward compatibility for non-generative models.

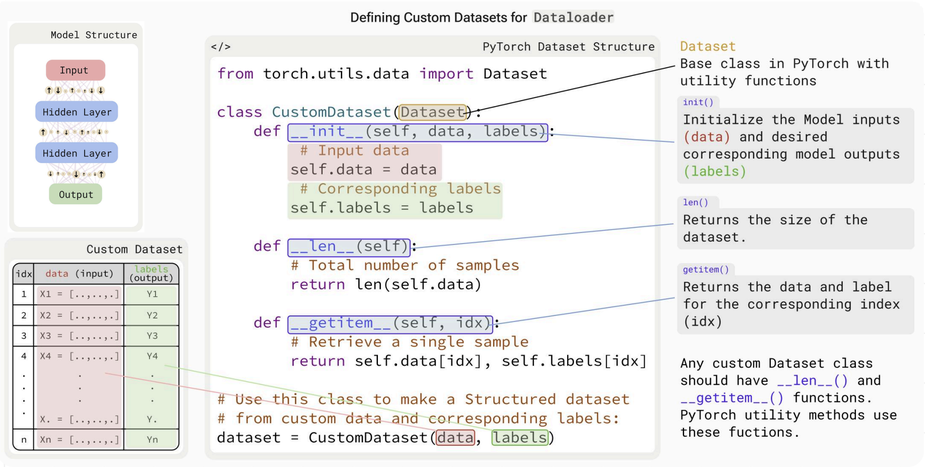
Thank you!

- ⭐ **Recognized as a Top contributor to the PyTorch Docathon 2025**

- Updated **new changes** to APIs in Automatic Mixed Precision and ONNX unified model format.

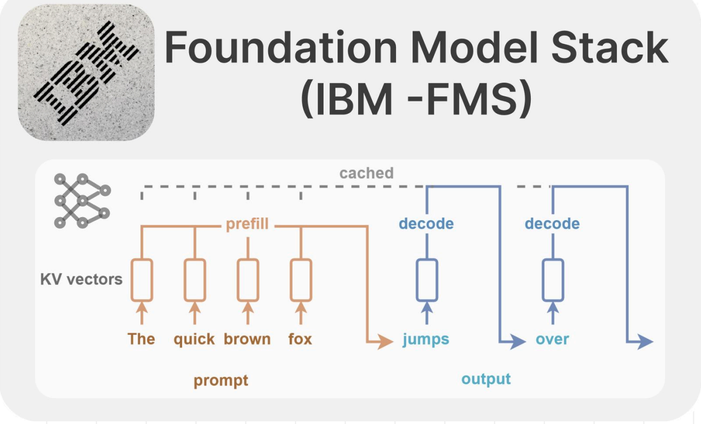
- Debugged and converted documentation from rst to MyST format for **12 functions** in latest update of docs.

  - 6 Pull Requests Created (4 Merged) **800+ lines each**
  - Reviewed 8 Pull Requests



- **Designed and Created 7 PyTorch labs for PyTorch Foundation's Certification and Training Initiative**

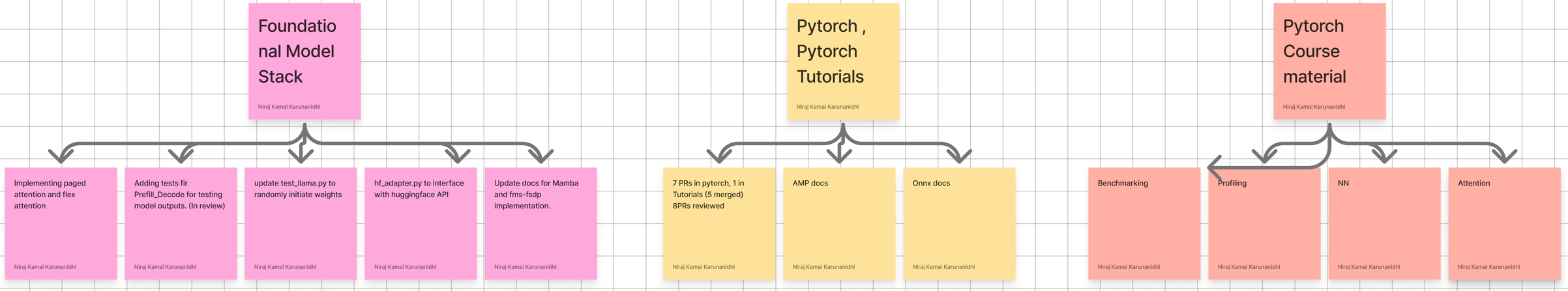
  - Intro to Pytorch
  - Building Neural Networks with PyTorch
  - Benchmarking Models
  - Leveraging Automatic Mixed Precision for training and inference
  - Activation Functions for Models
  - Performance Profiler for Models
  - Creating Neural Network Checkpoints



- **Added support for Decoder based LLM Model verification - for Prefill and Decode Stages.**

- Improved the handling of **generative models** (e.g., causal language models like LLaMA) while maintaining backward compatibility for non-generative models.

Niraj Kamal Karunanidhi
MS in Computer Science
IBM Open Source AI Intern
Georgia Institute of Technology



AI Research Summer Internship

Niraj Kamal Karunanidhi