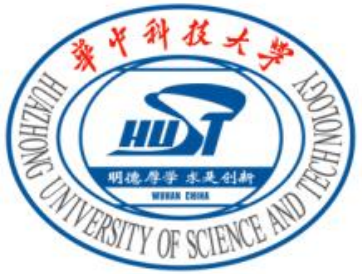


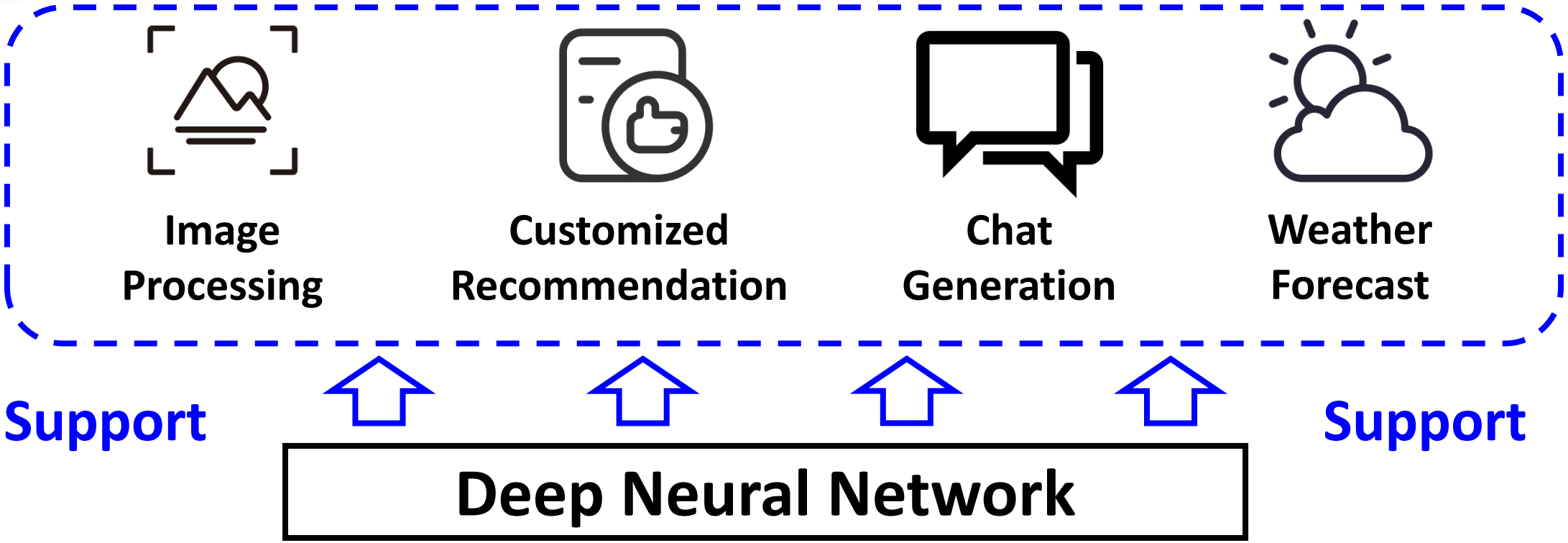
A Cost-Efficient Failure-Tolerant Scheme for Distributed DNN Training

Menglei Chen, Yu Hua, Rong Bai, Jianming Huang
Huazhong University of Science and Technology, China

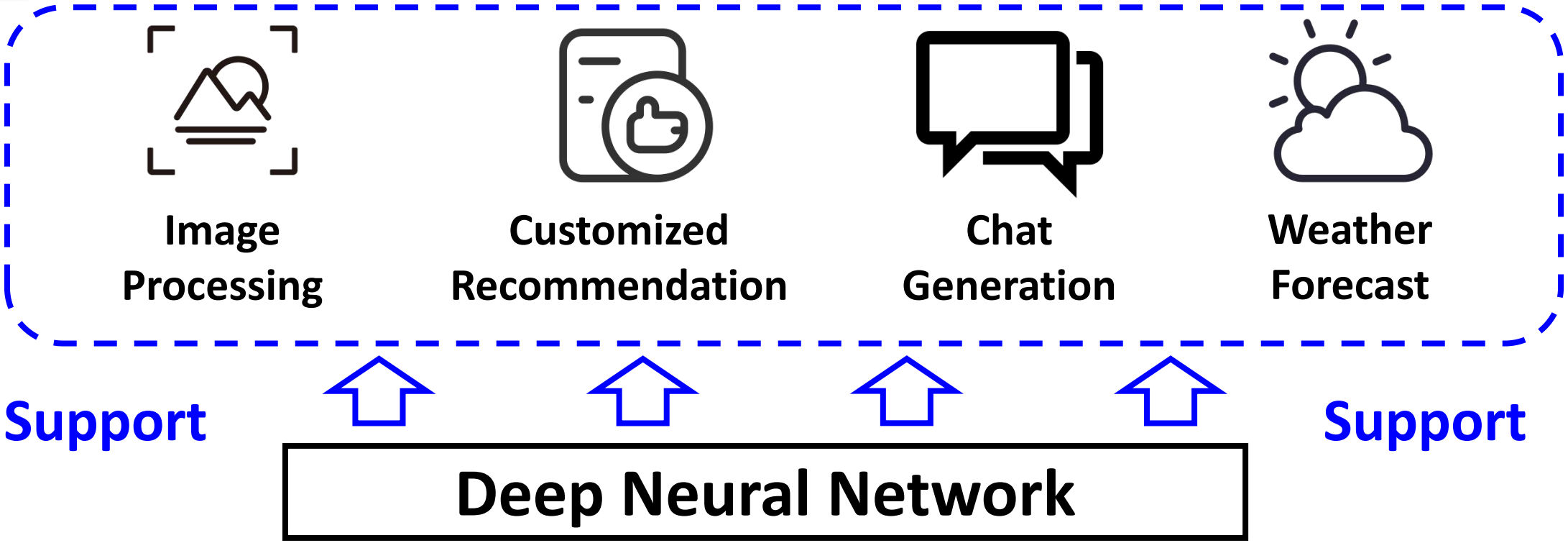
Deep Neural Network (DNN)

Deep Neural Network

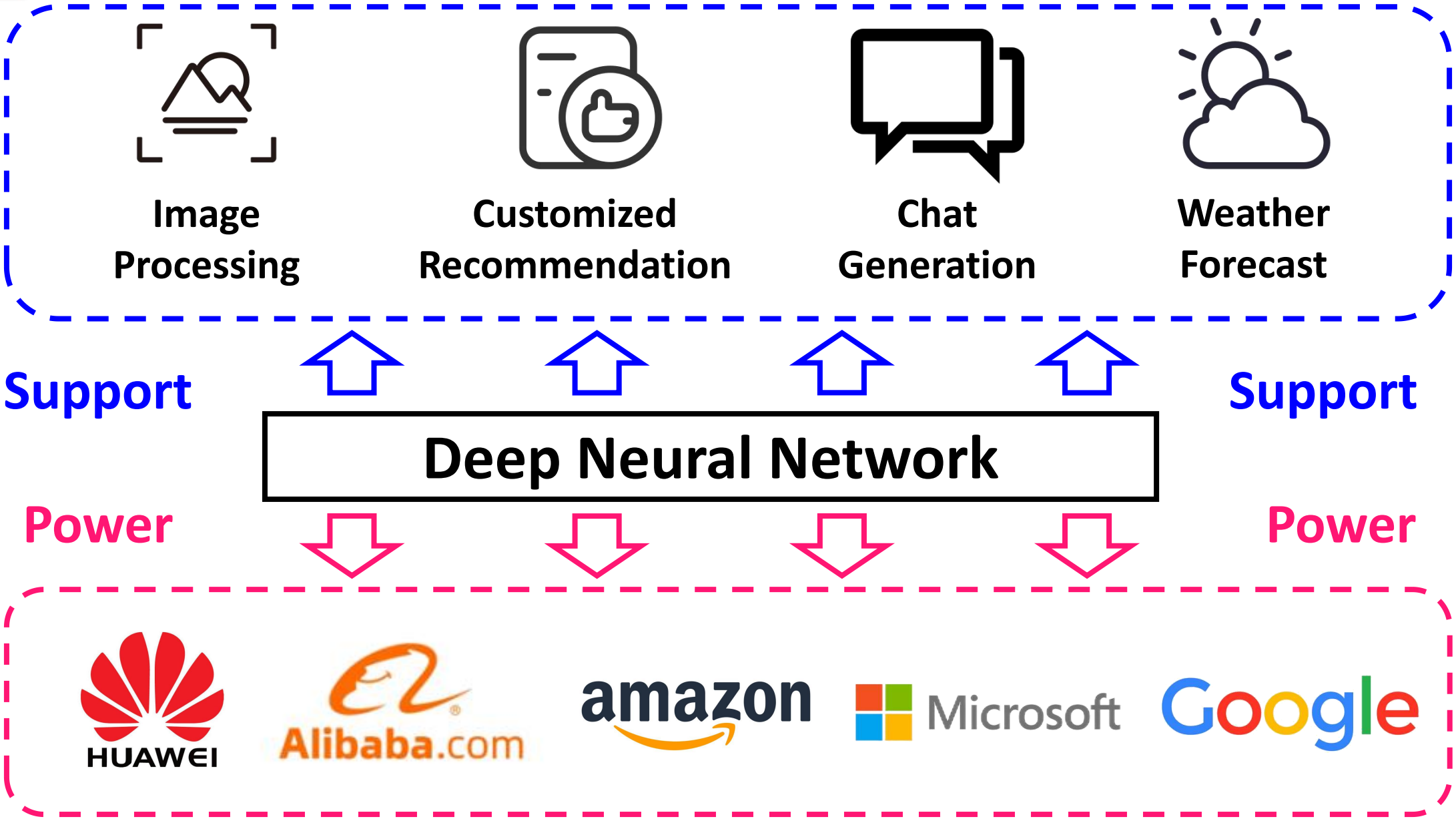
Deep Neural Network (DNN)



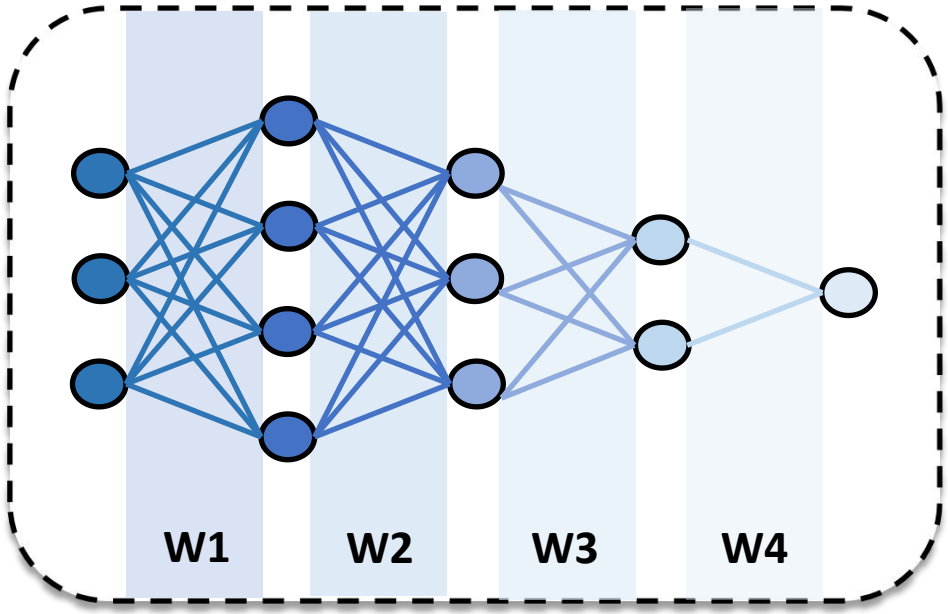
Deep Neural Network (DNN)



Deep Neural Network (DNN)

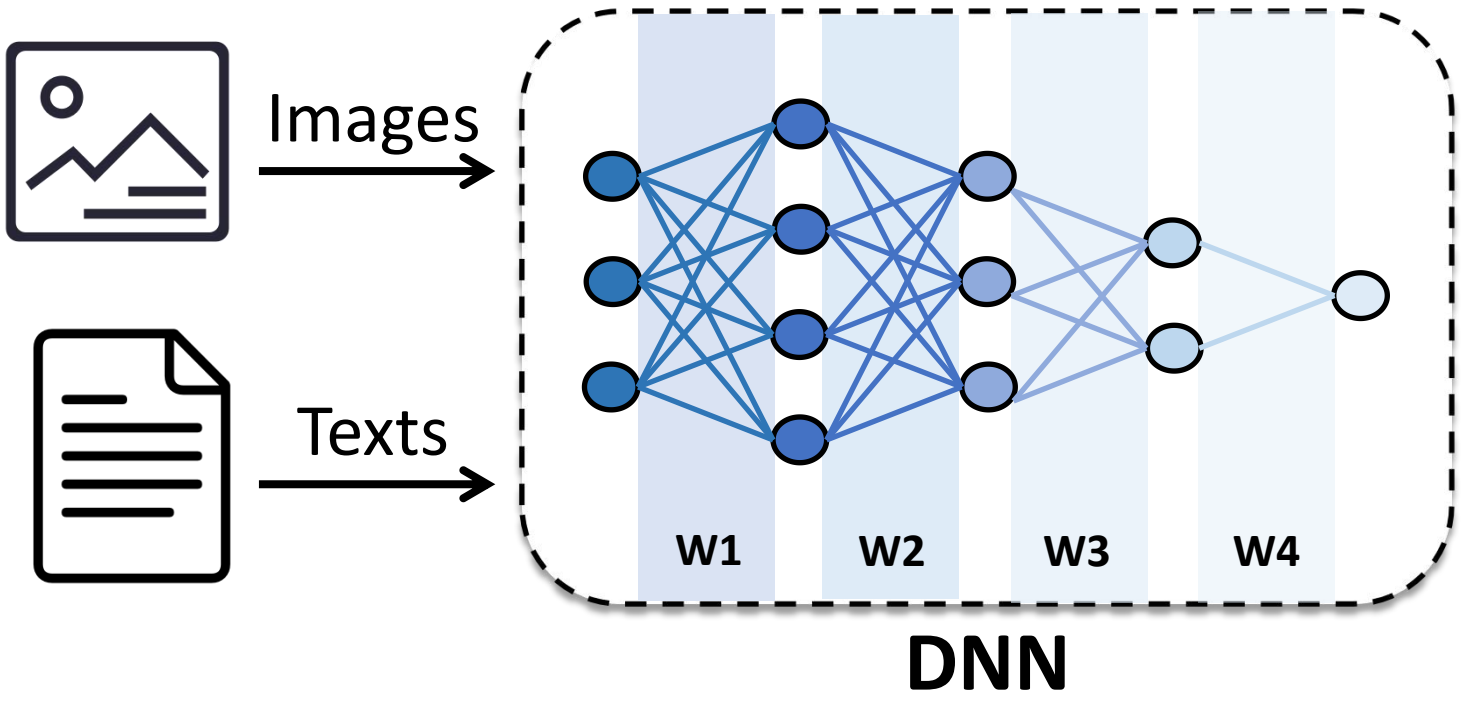


DNN Training

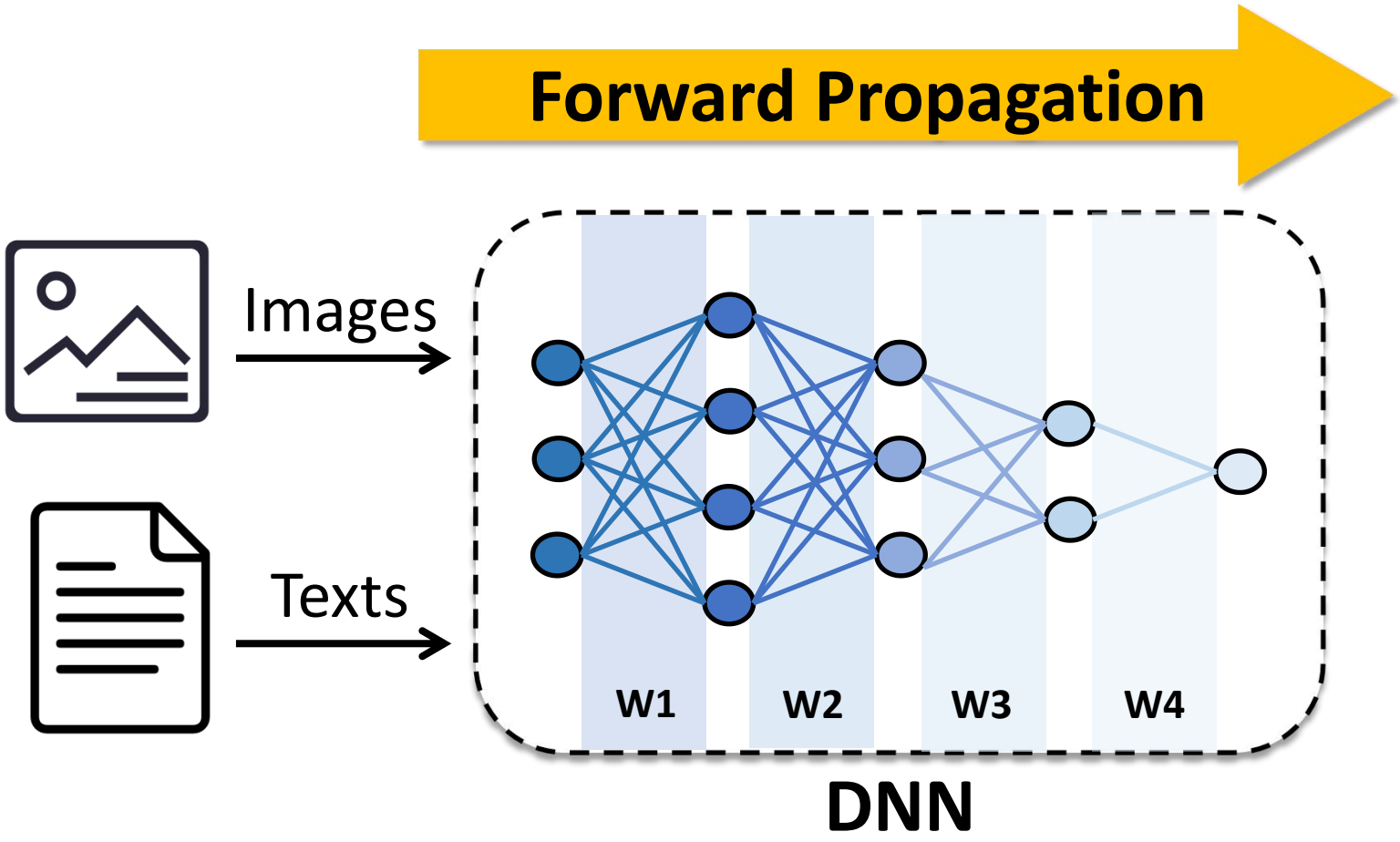


DNN

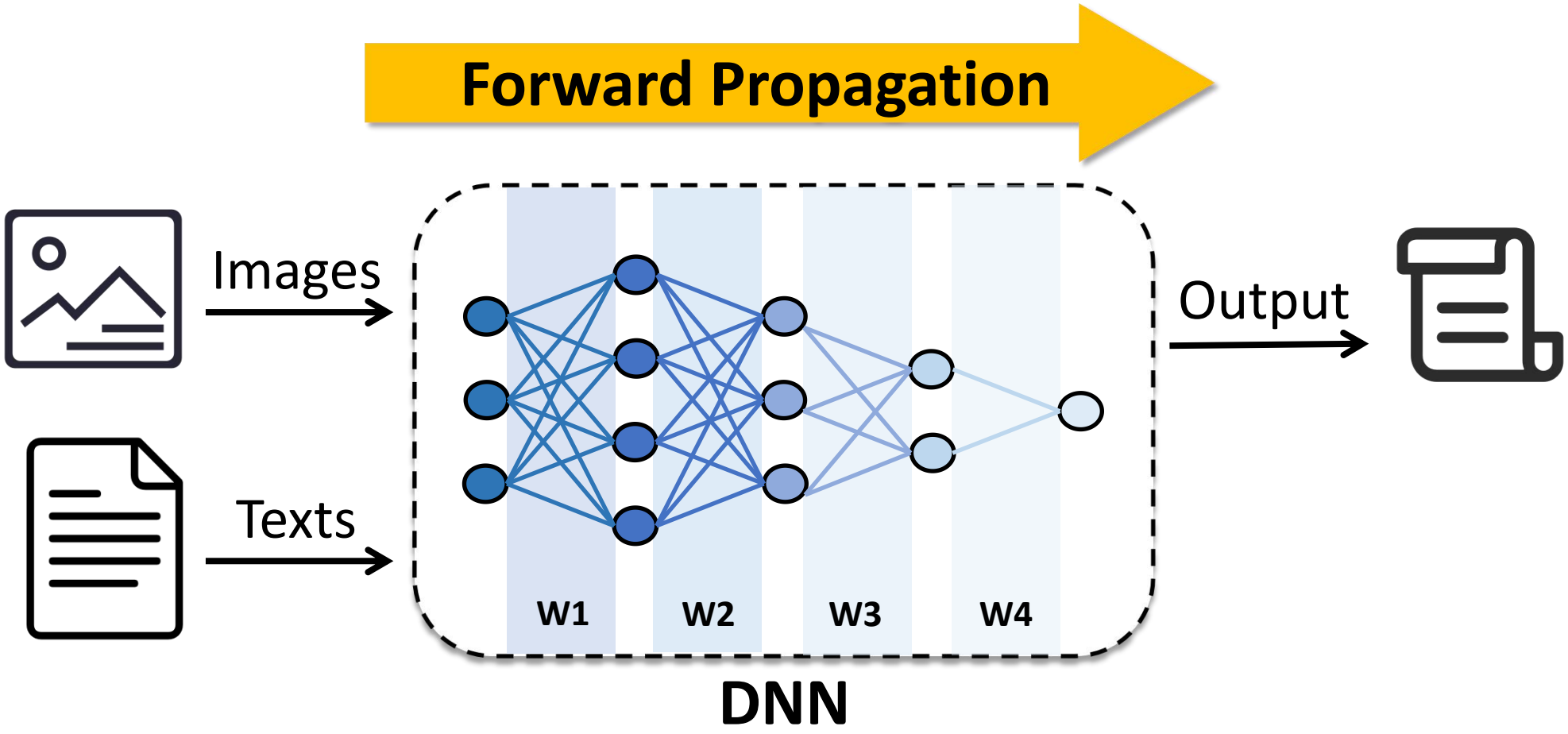
DNN Training



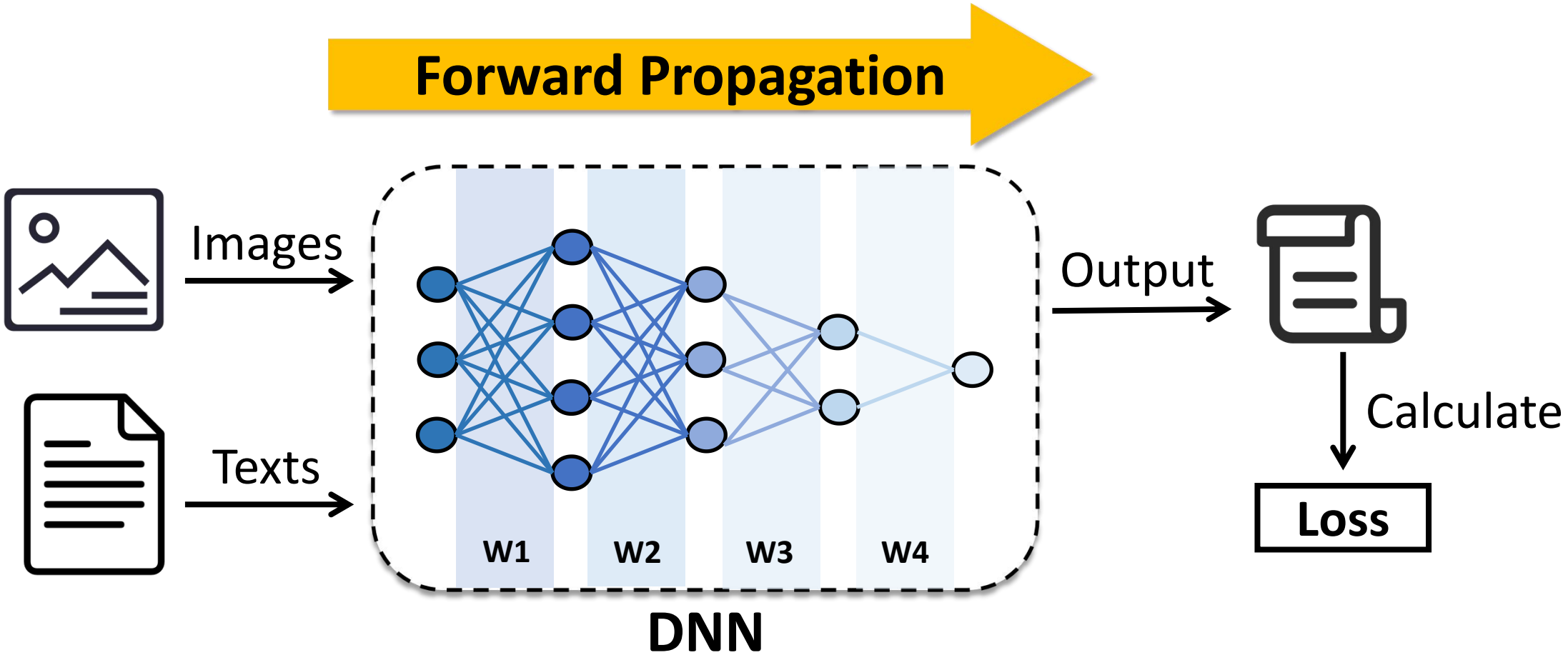
DNN Training



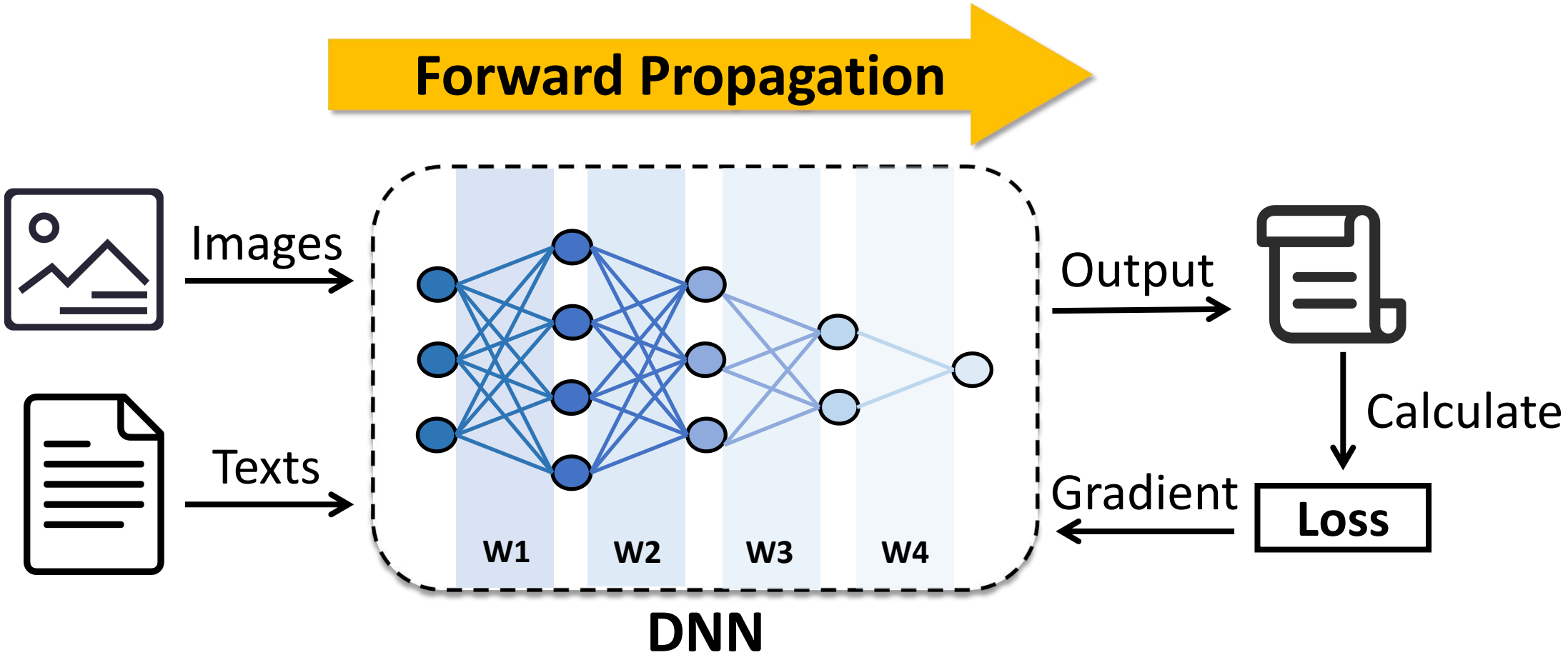
DNN Training



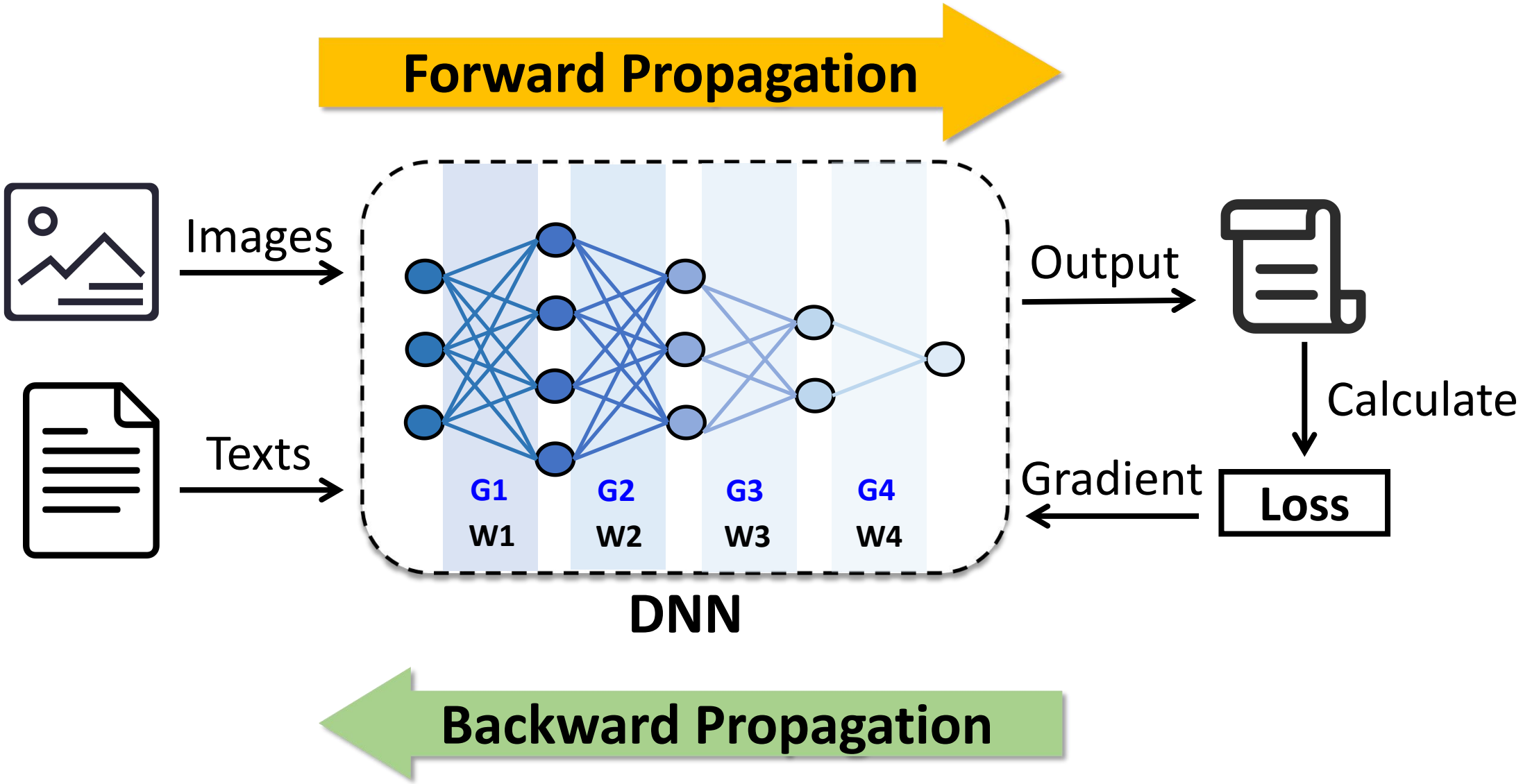
DNN Training



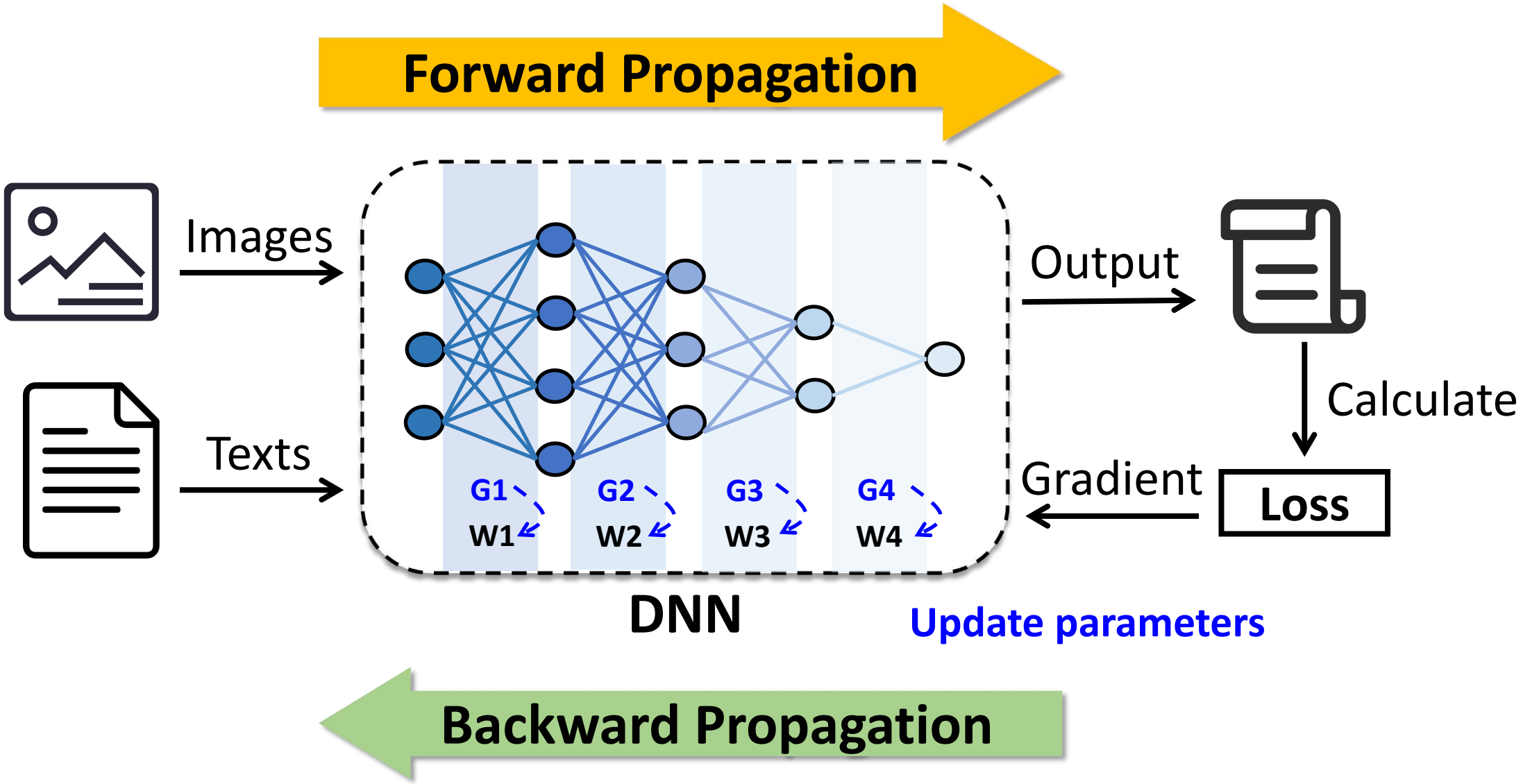
DNN Training



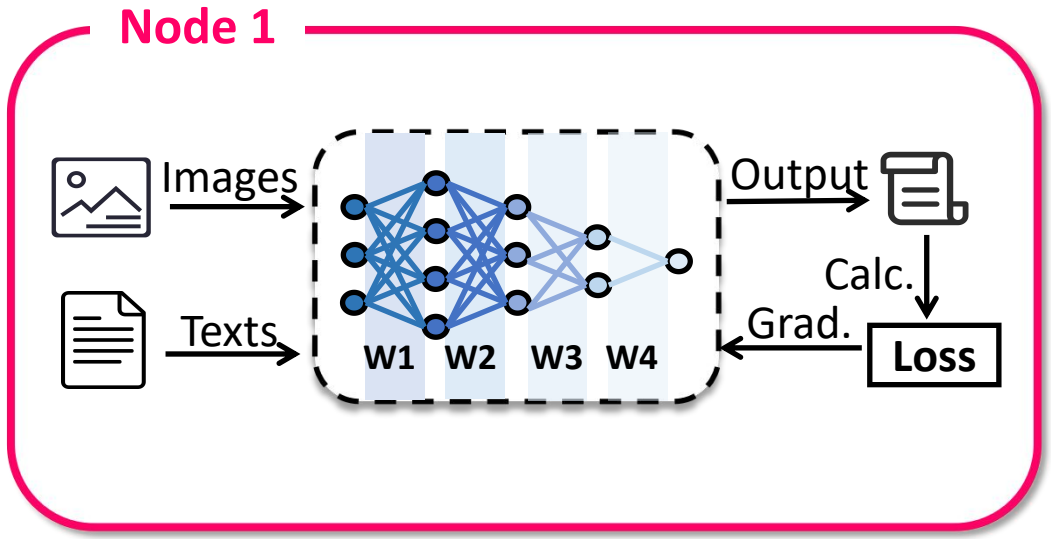
DNN Training



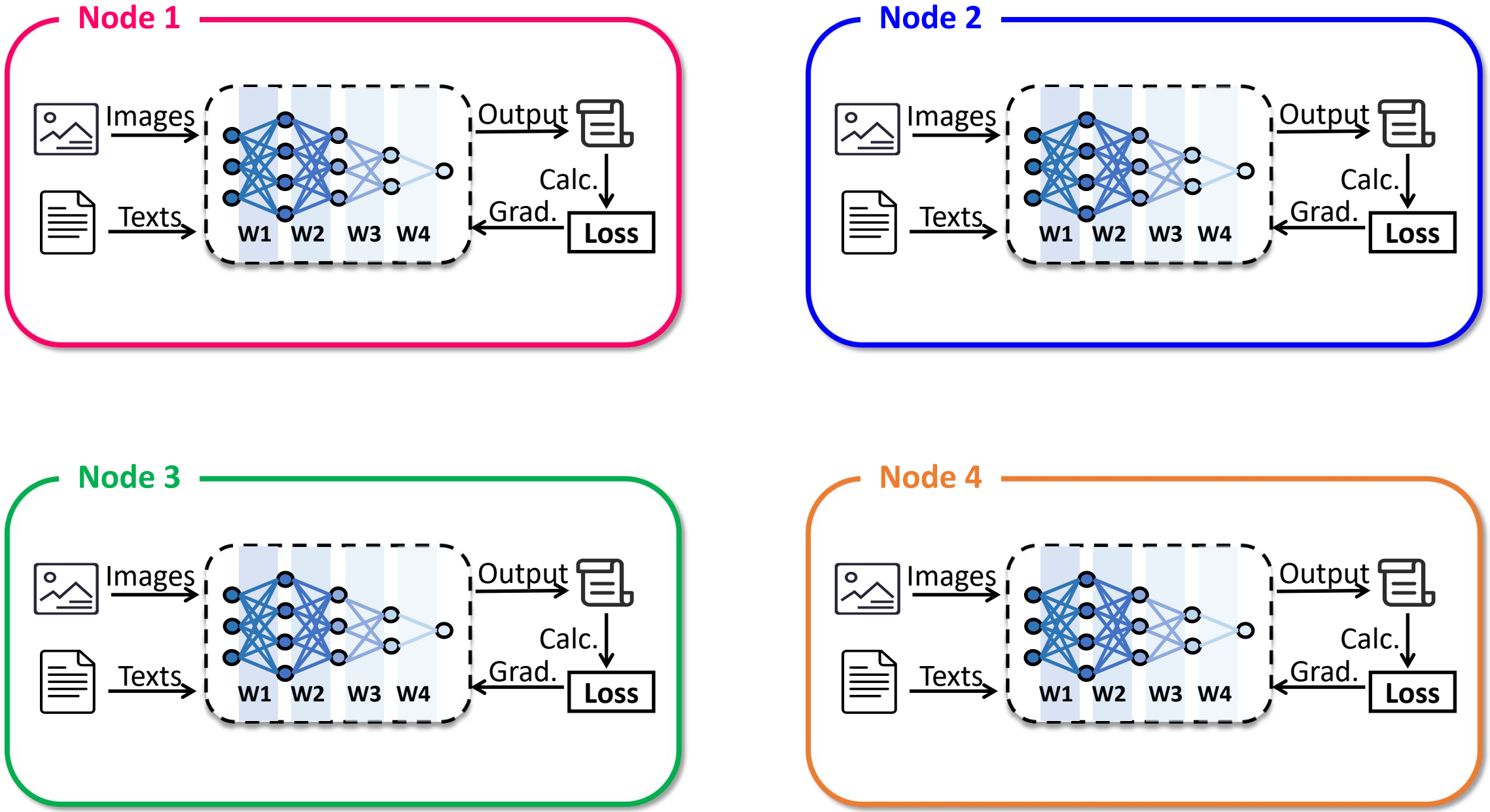
DNN Training



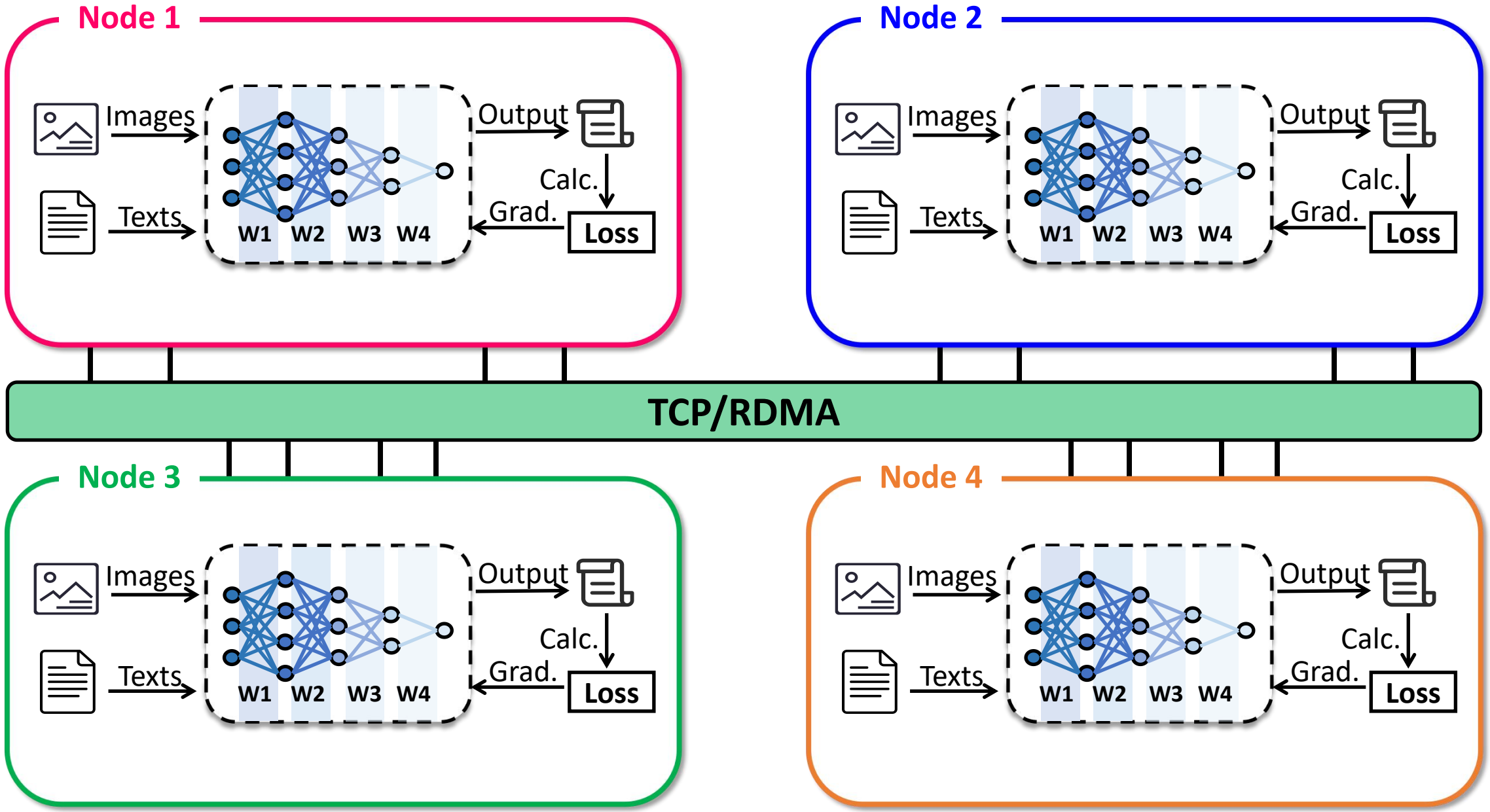
Distributed DNN Training



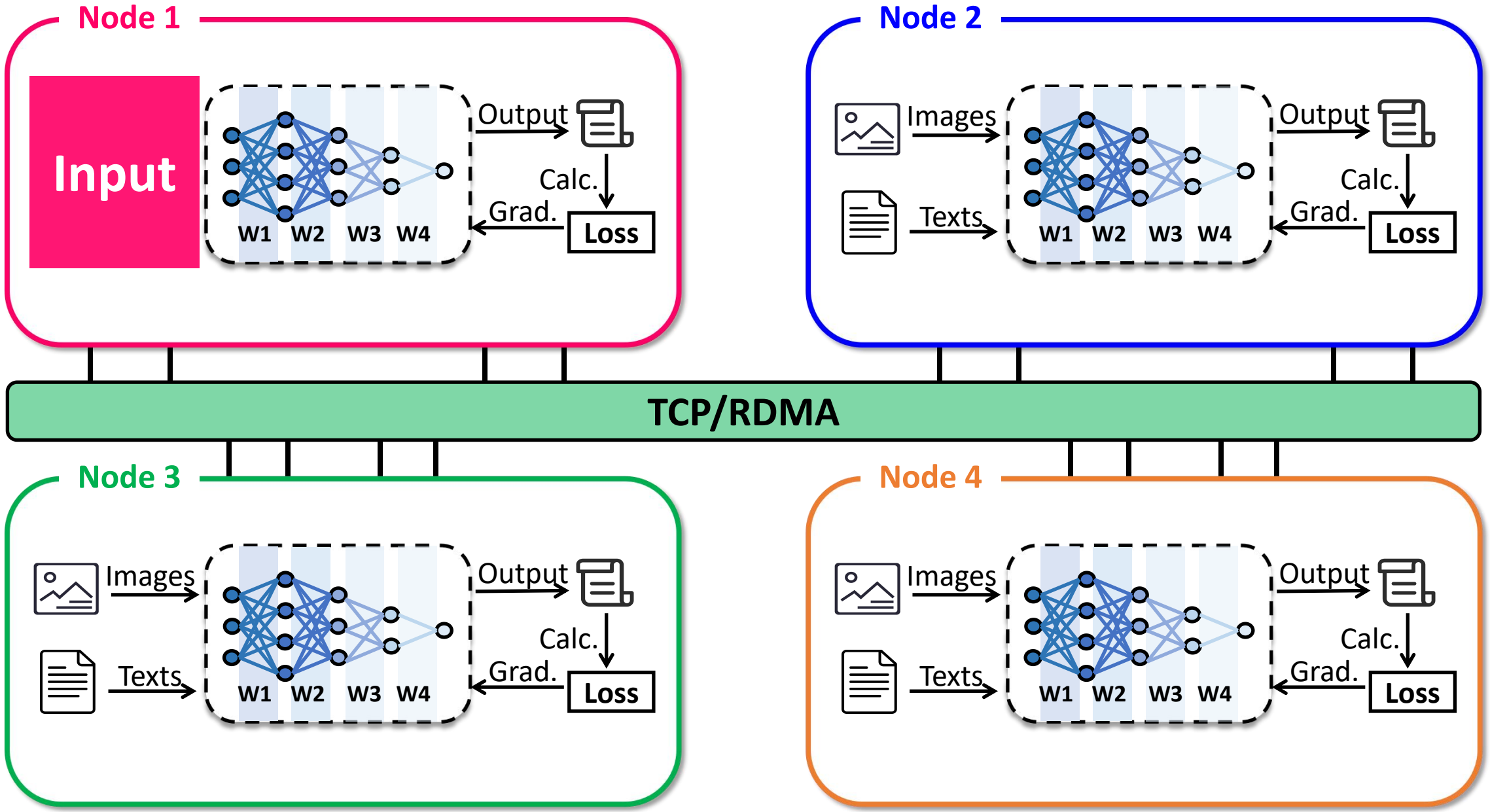
Distributed DNN Training



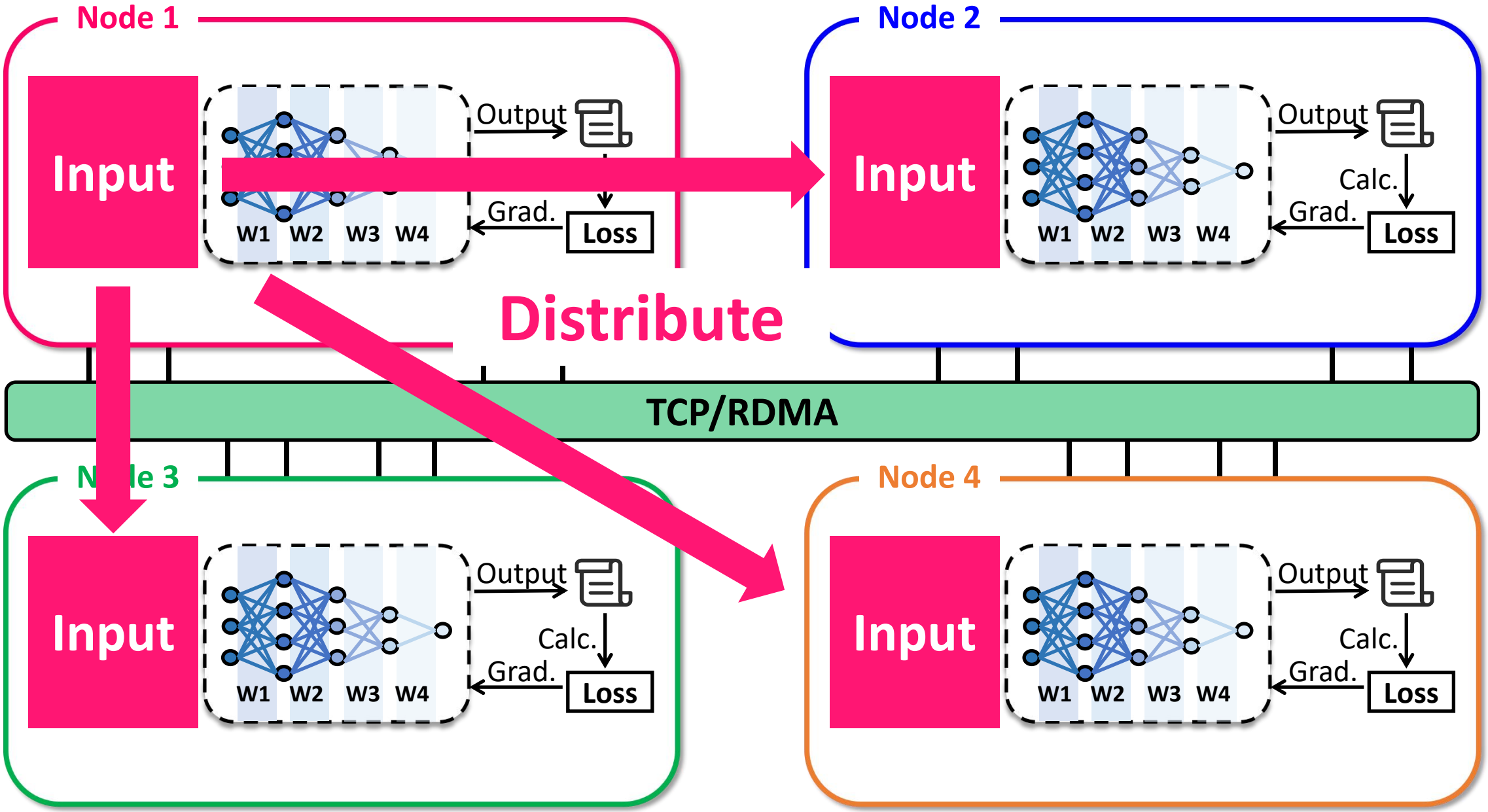
Distributed DNN Training



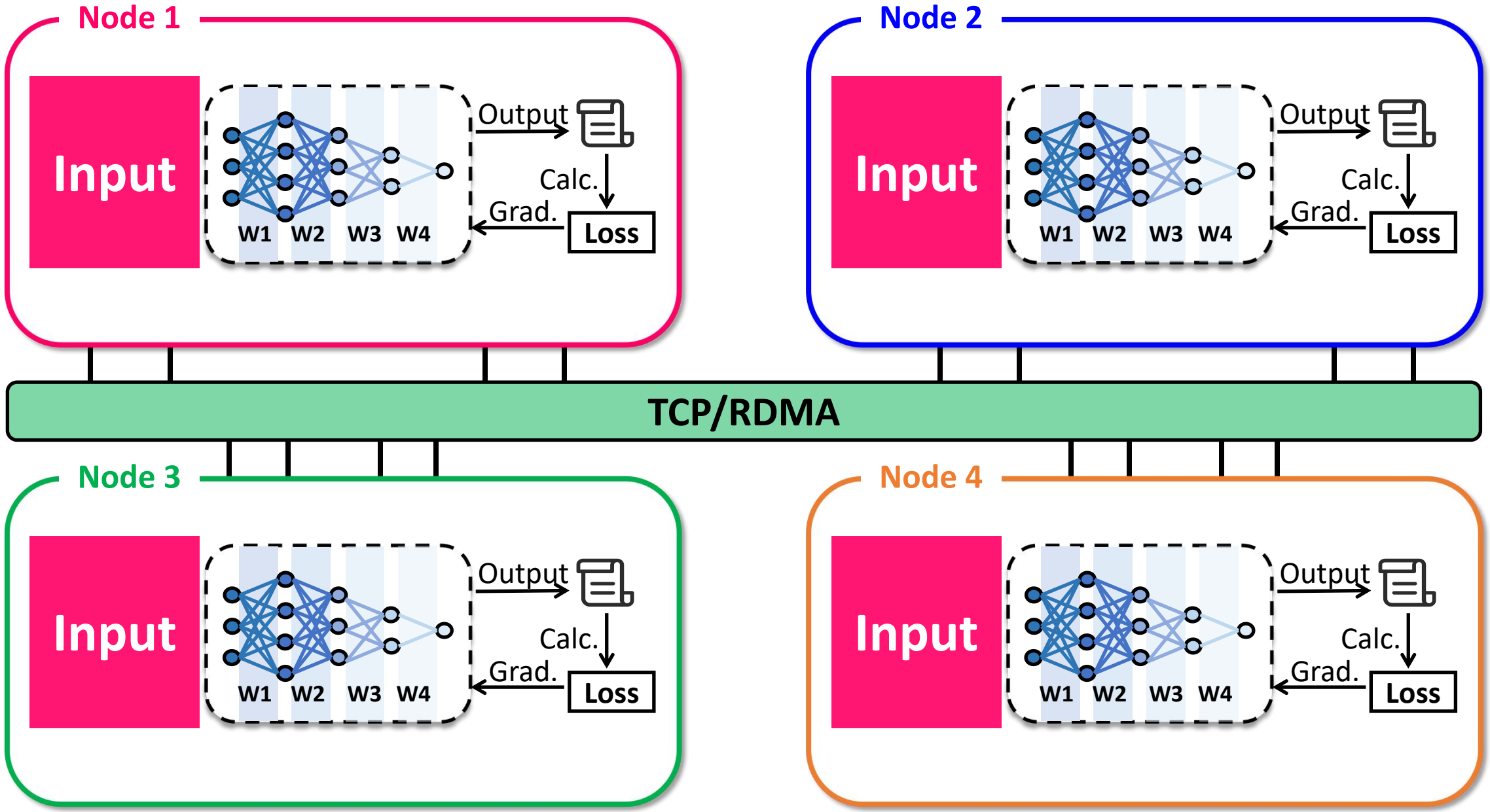
Distributed DNN Training



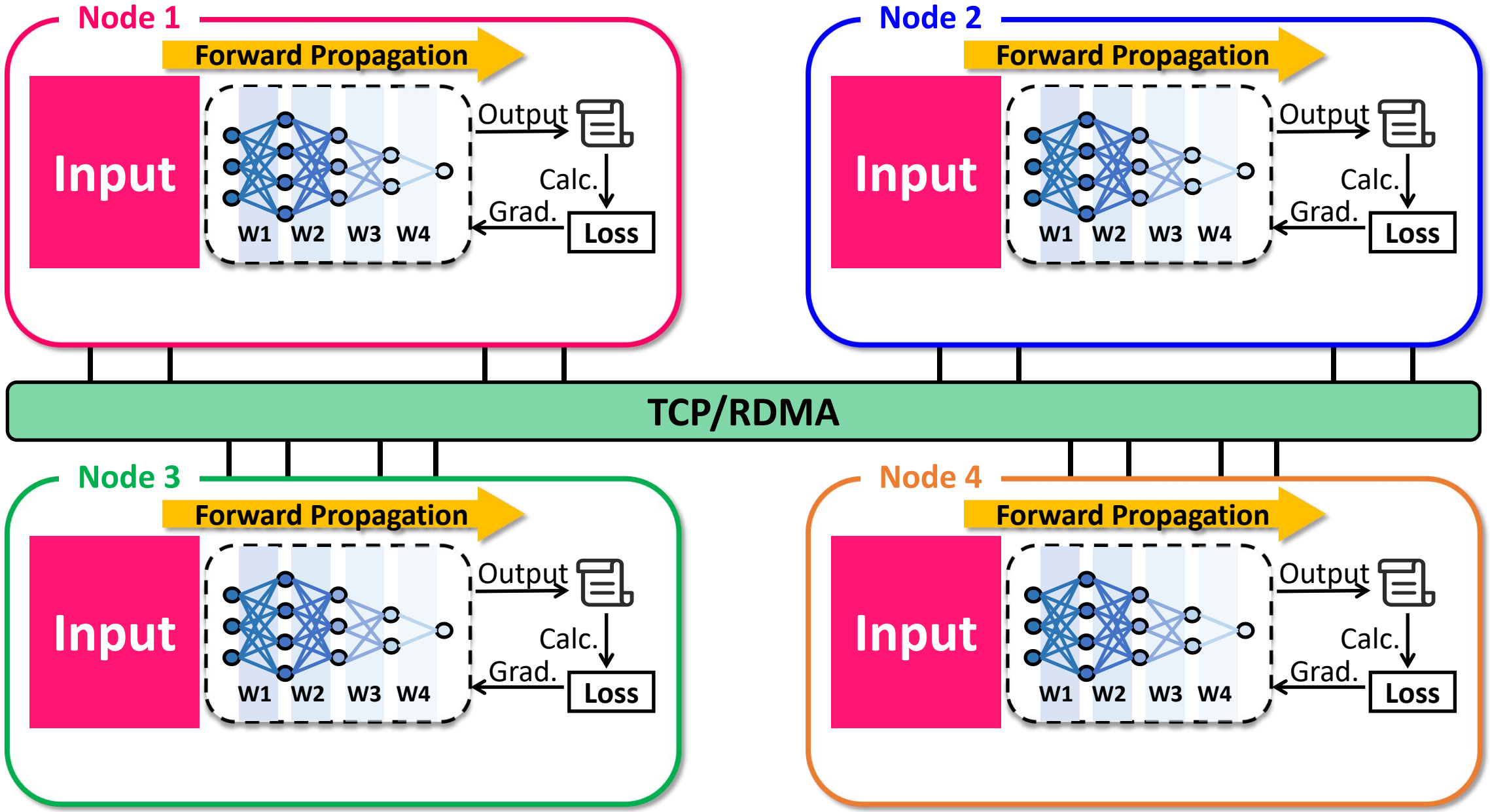
Distributed DNN Training



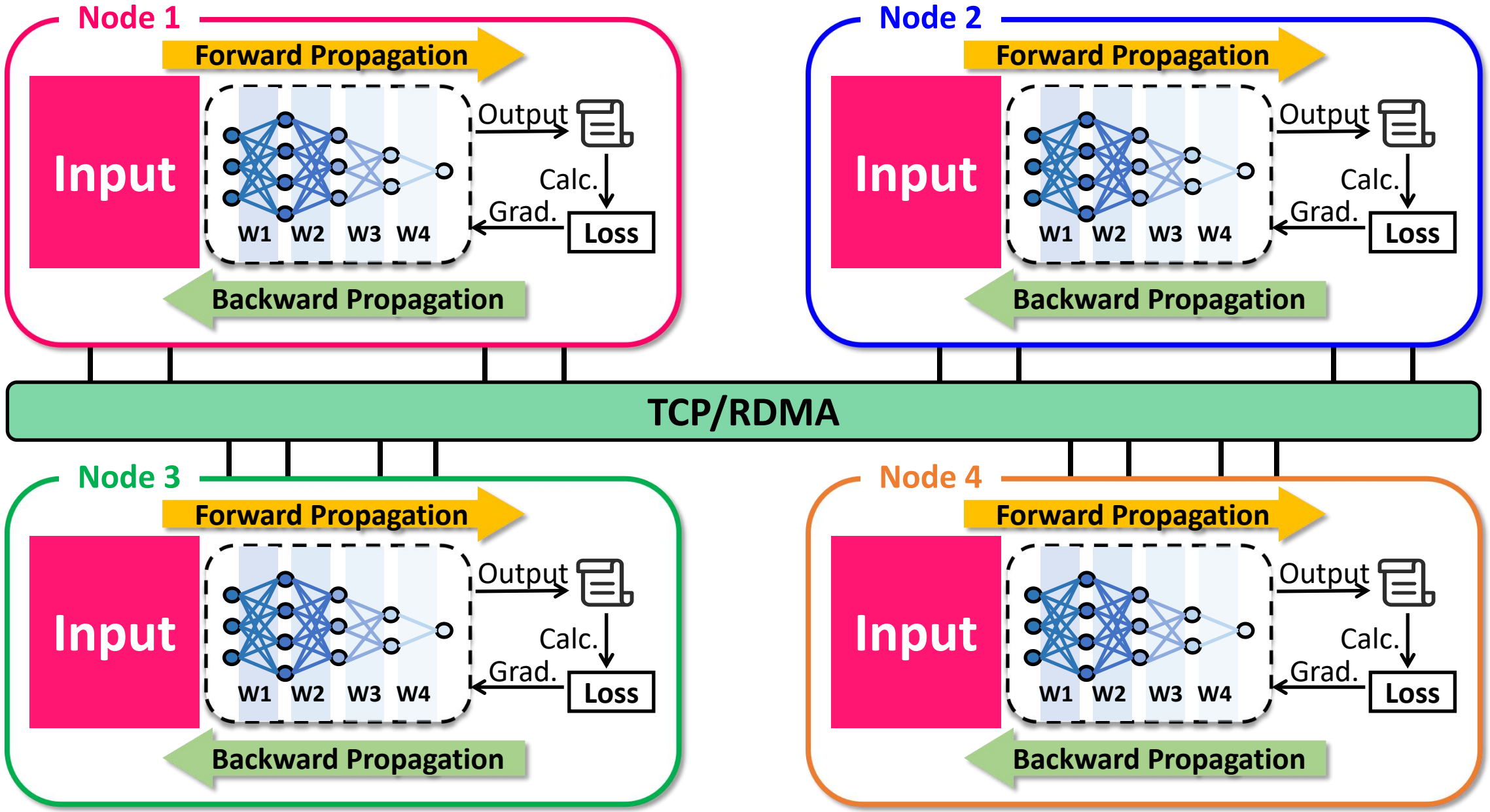
Distributed DNN Training



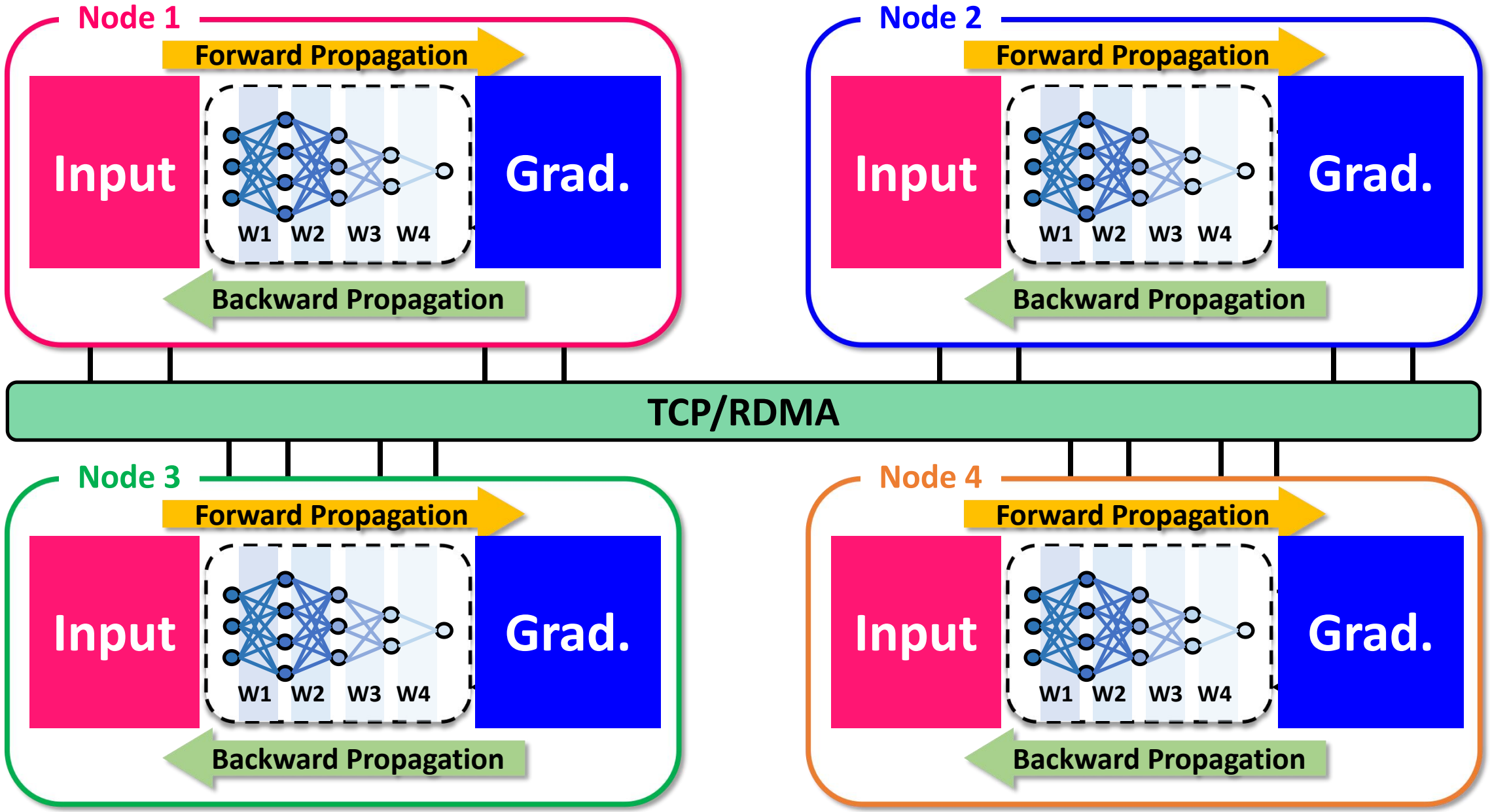
Distributed DNN Training



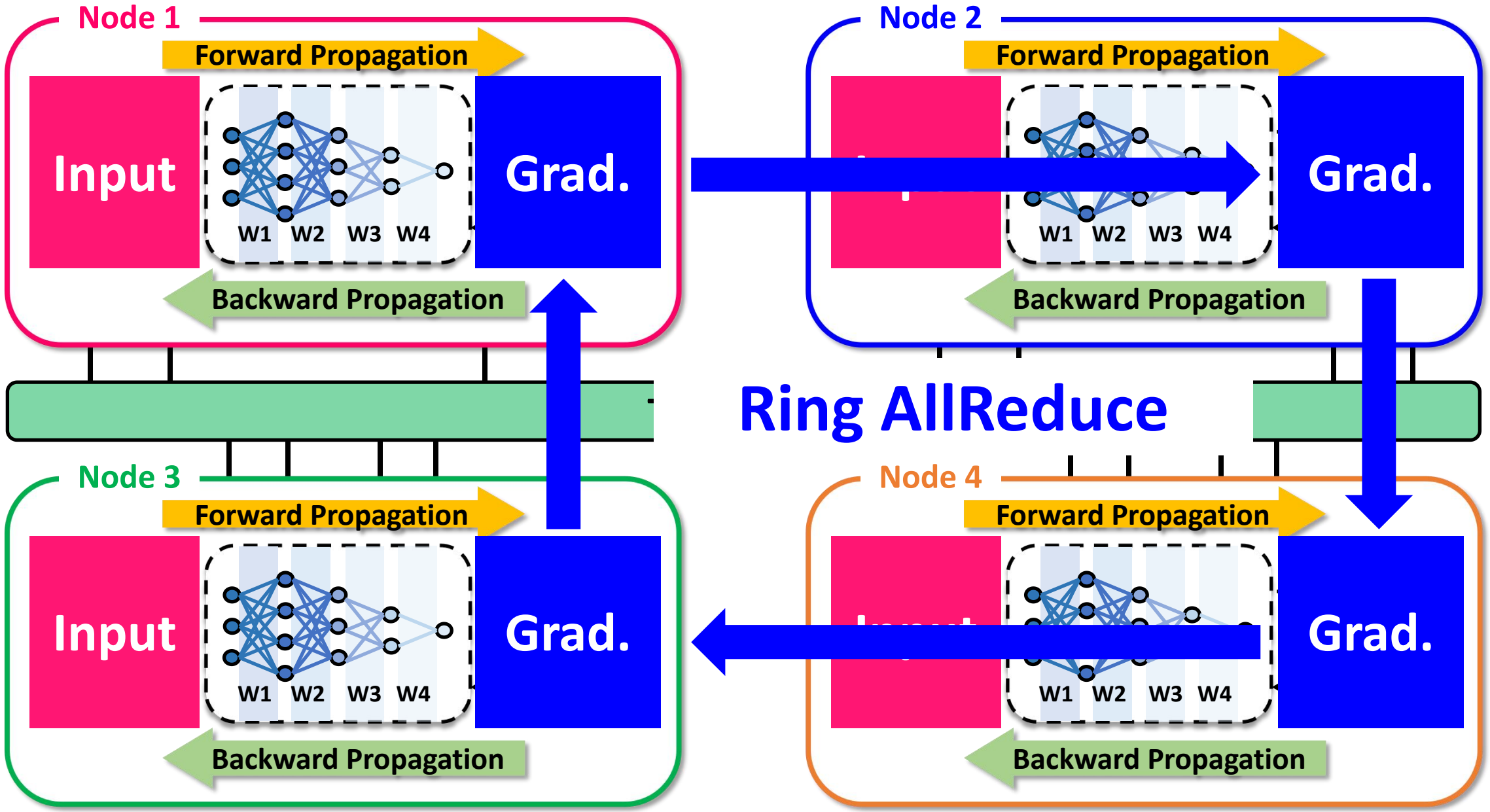
Distributed DNN Training



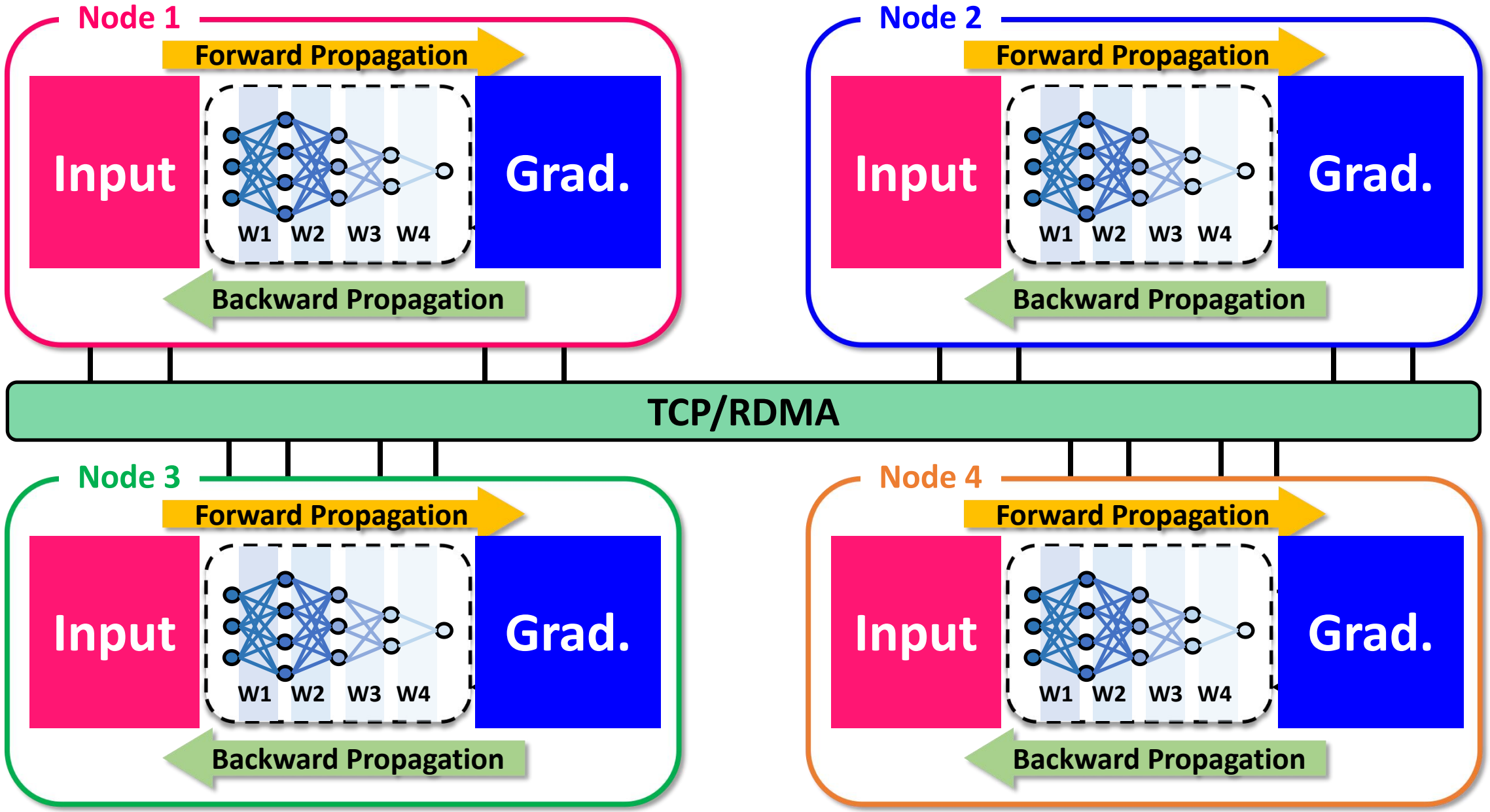
Distributed DNN Training



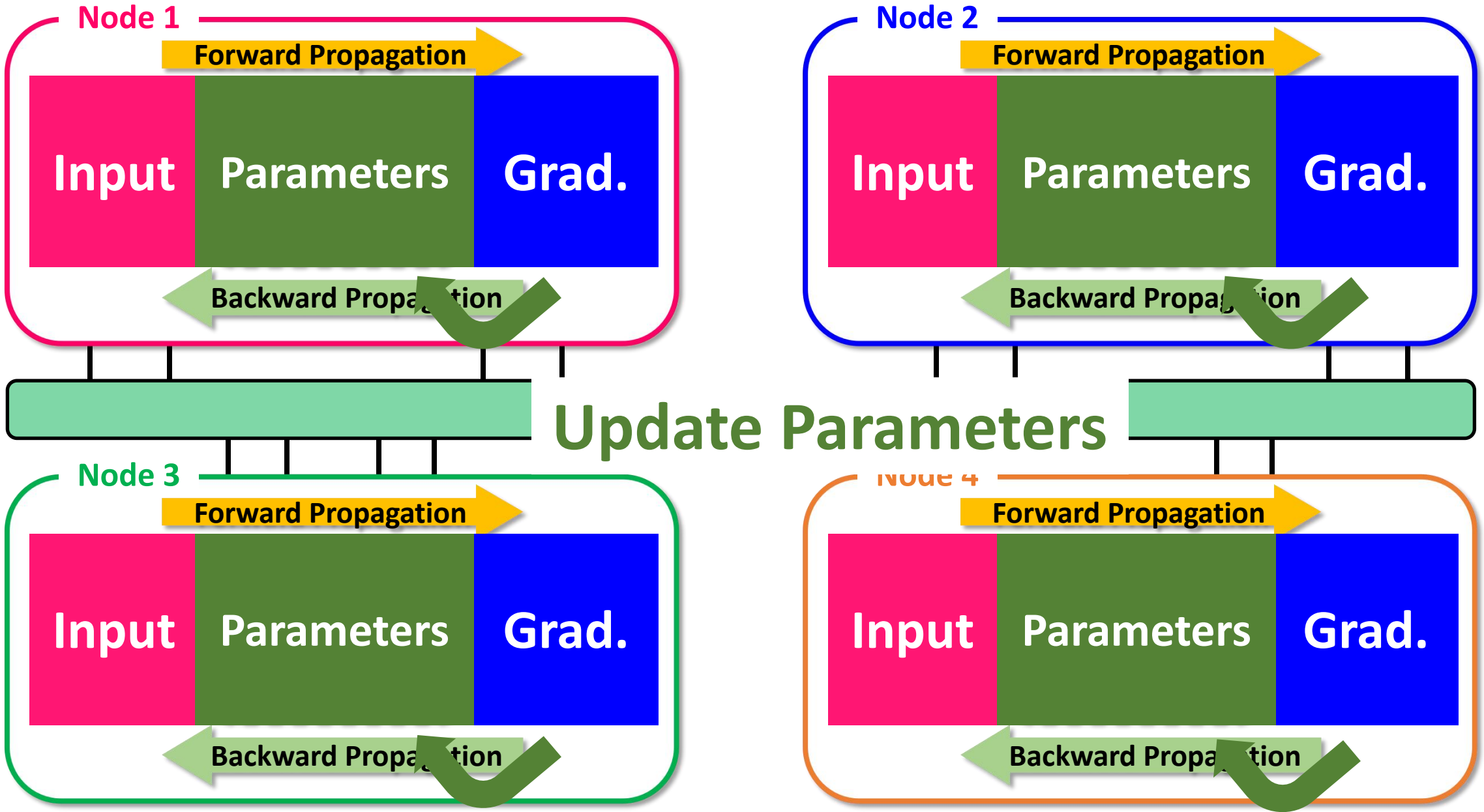
Distributed DNN Training



Distributed DNN Training



Distributed DNN Training



The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

Training GPT-3

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

→ **Thousands of A100 GPUs**

Training GPT-3

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

Thousands of A100 GPUs

Training GPT-3

Several months

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

Training GPT-3 → **Thousands of A100 GPUs**

→ **Several months**

→ **5+ Million dollars**

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

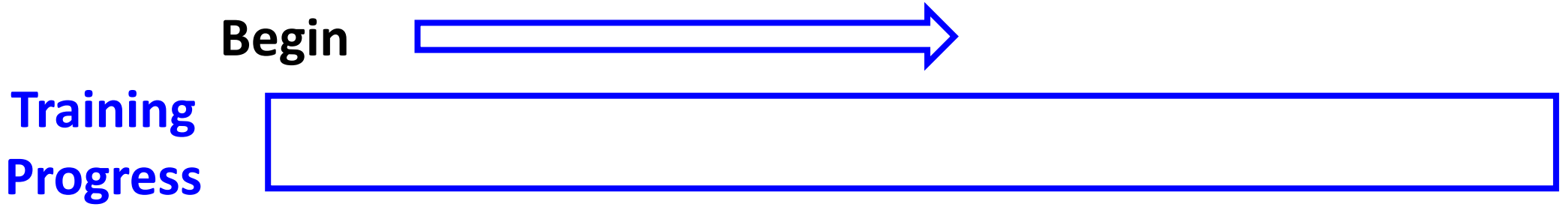
The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

Training Progress

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

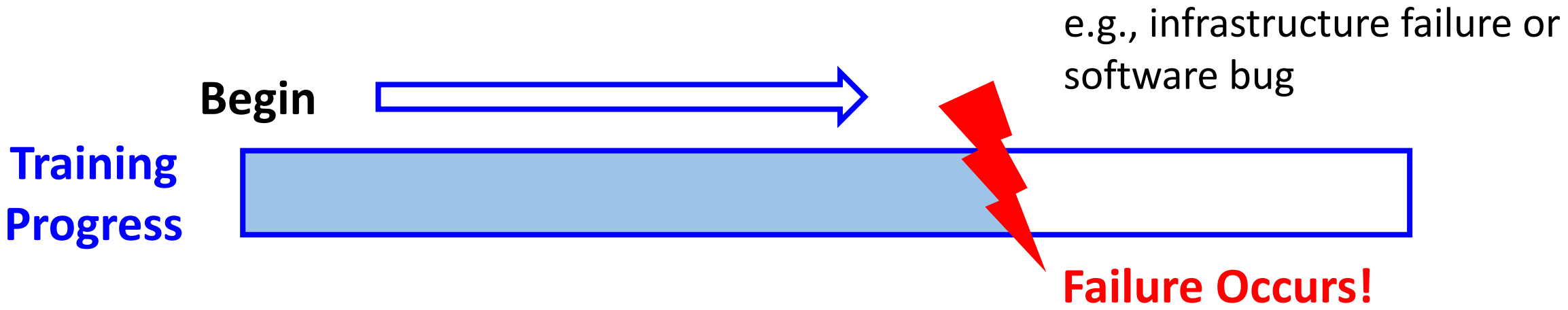


The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

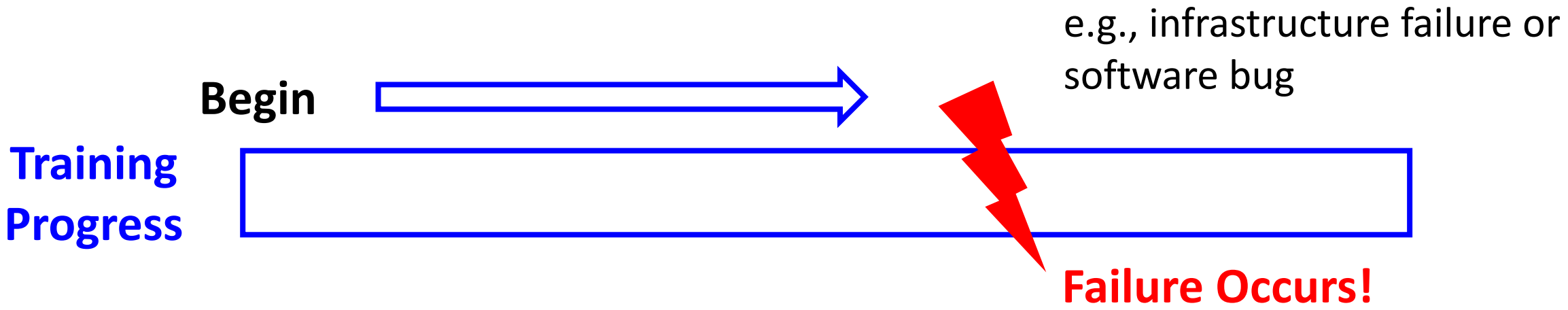
The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**



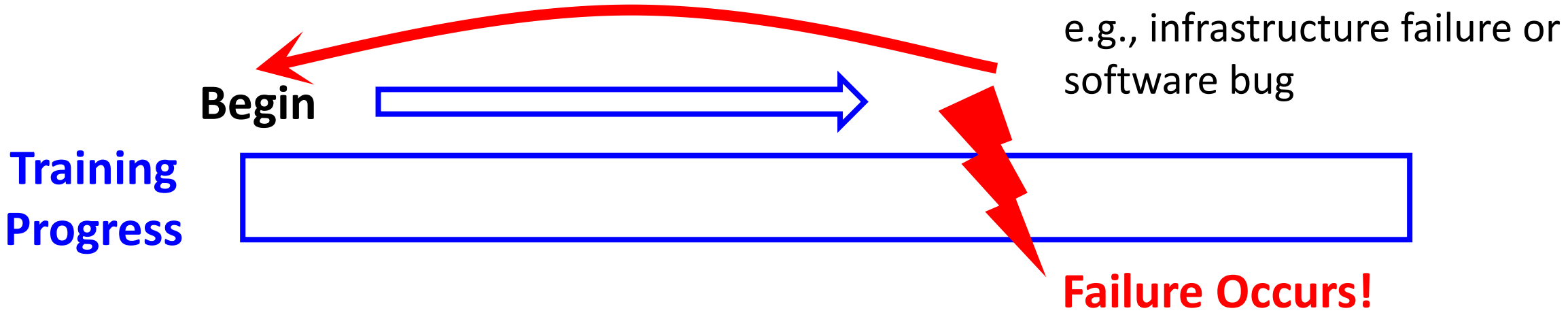
The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**



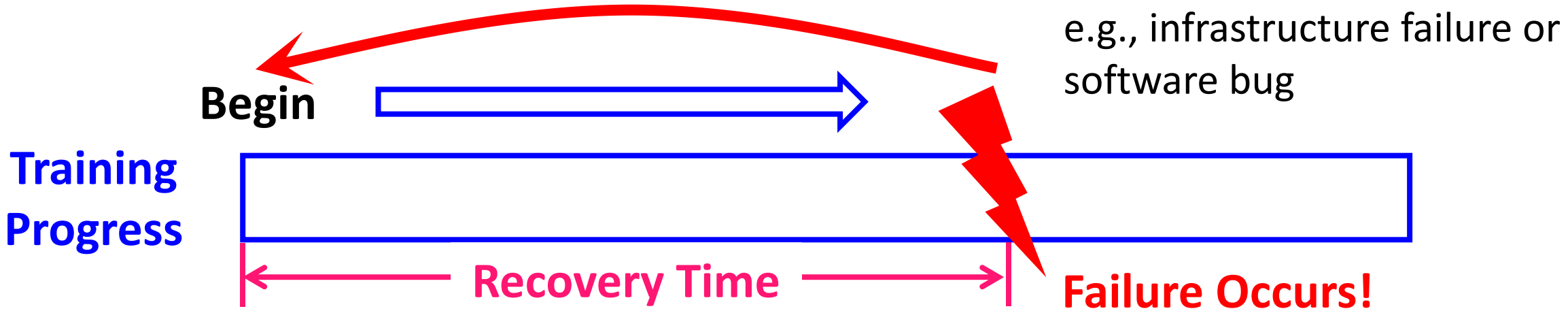
The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**



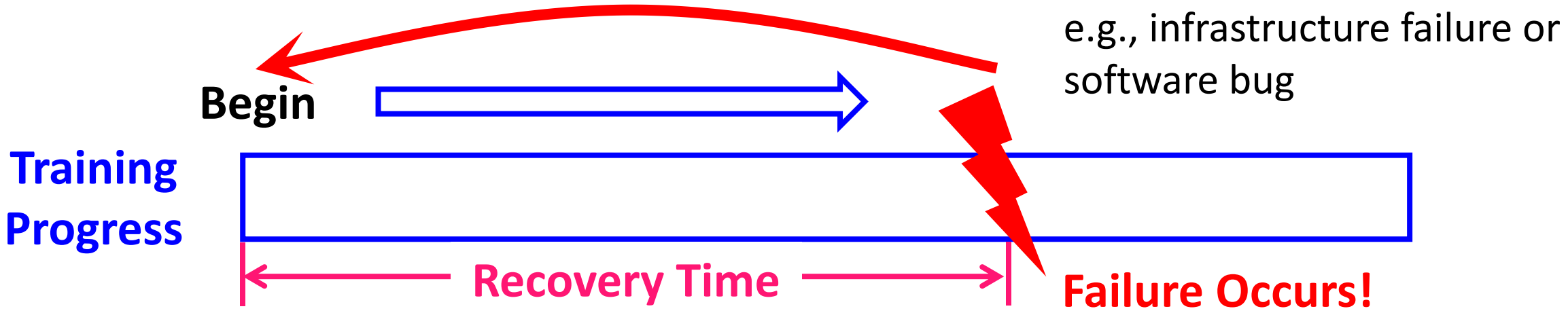
The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**



The importance of Failure Tolerance

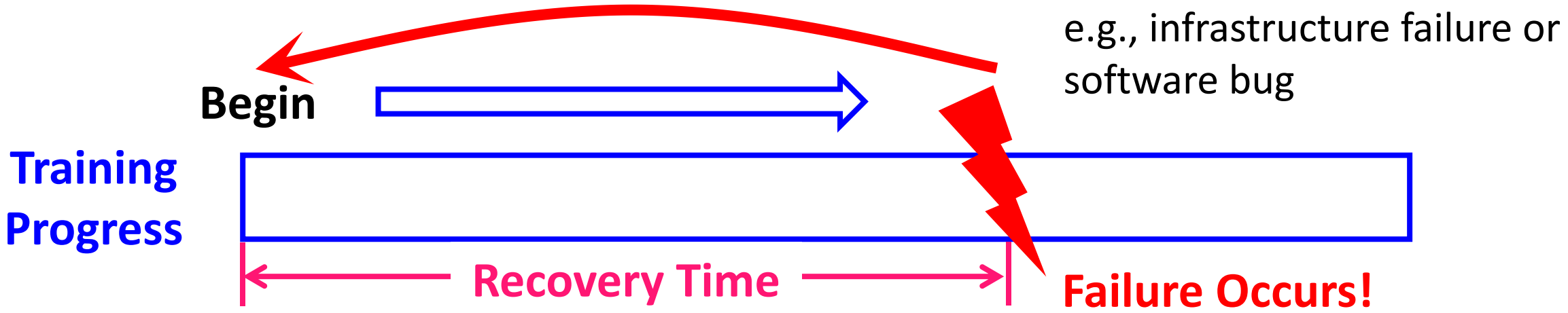
- DNN training is **time-consuming** and **expensive**



- **Checkpointing** is an efficient way to **ensure failure tolerance**

The importance of Failure Tolerance

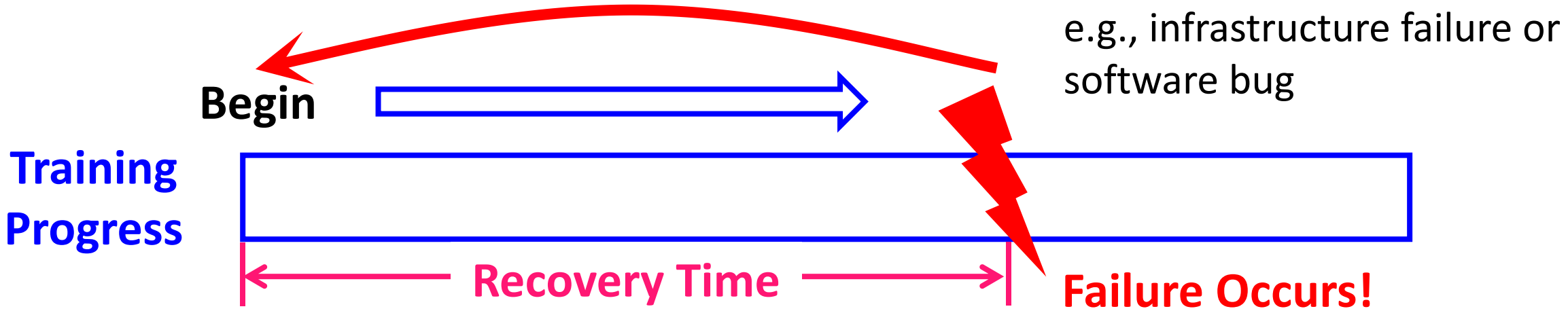
- DNN training is **time-consuming** and **expensive**



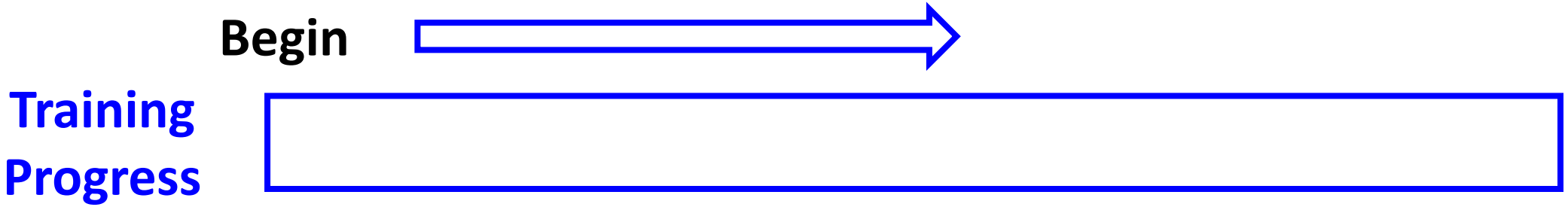
- **Checkpointing** is an efficient way to **ensure failure tolerance**

The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

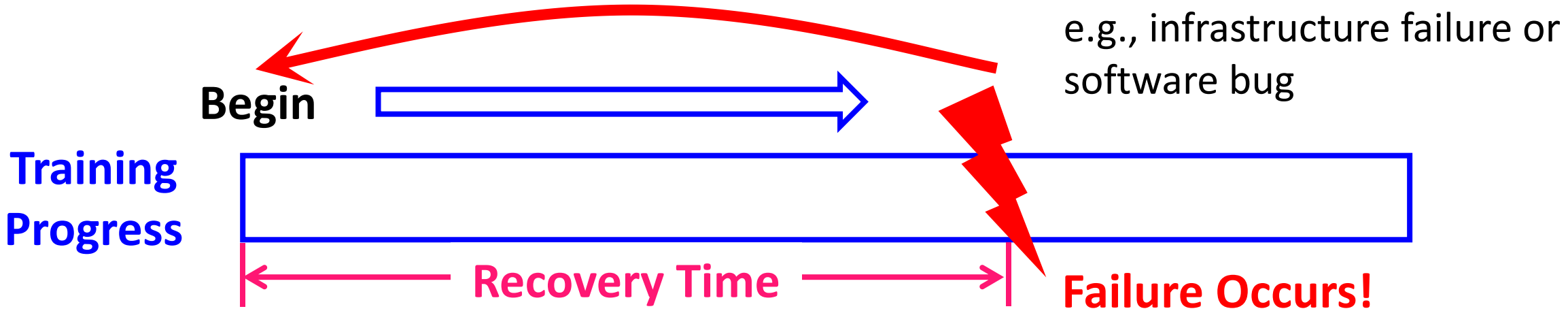


- **Checkpointing** is an efficient way to **ensure failure tolerance**

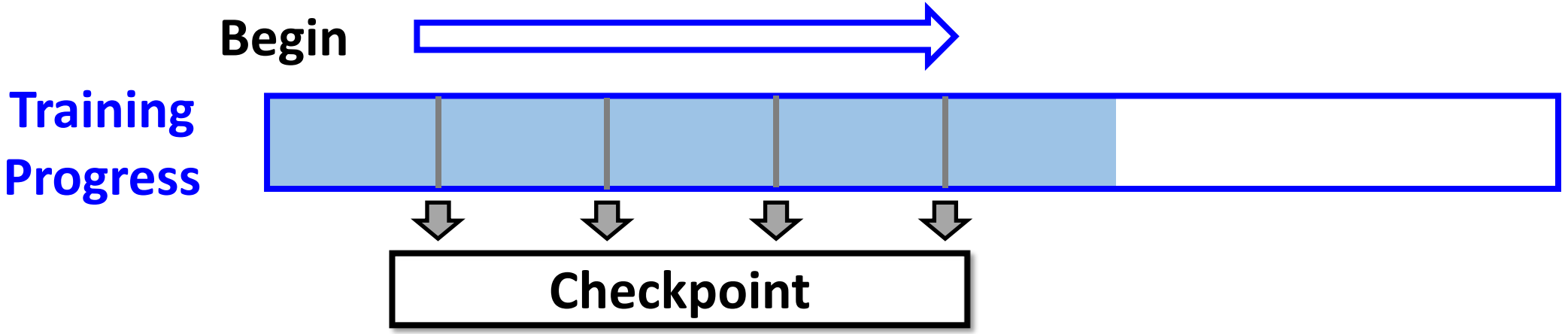


The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

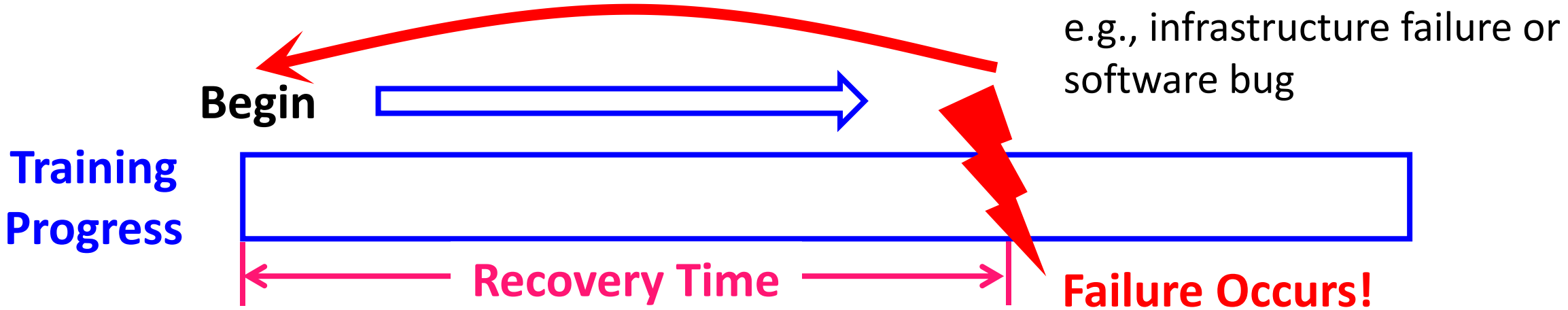


- **Checkpointing** is an efficient way to **ensure failure tolerance**

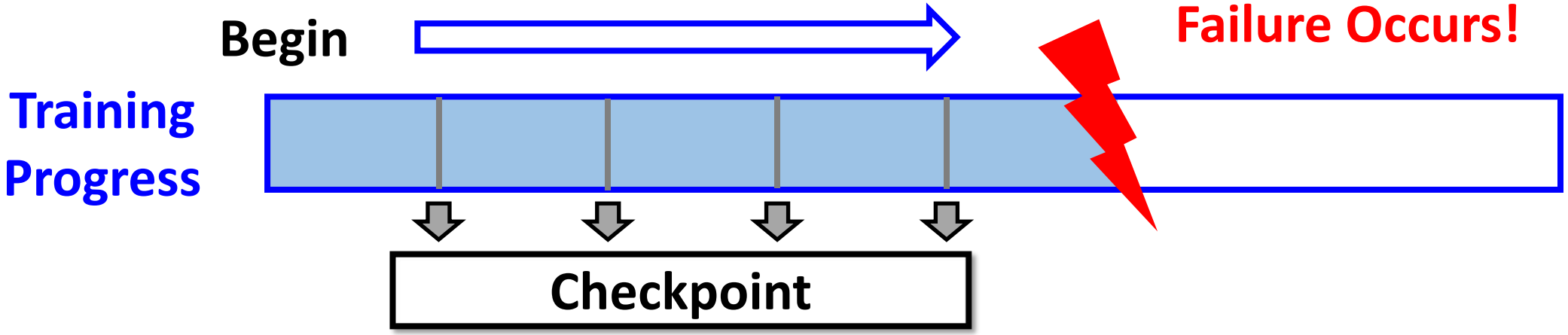


The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

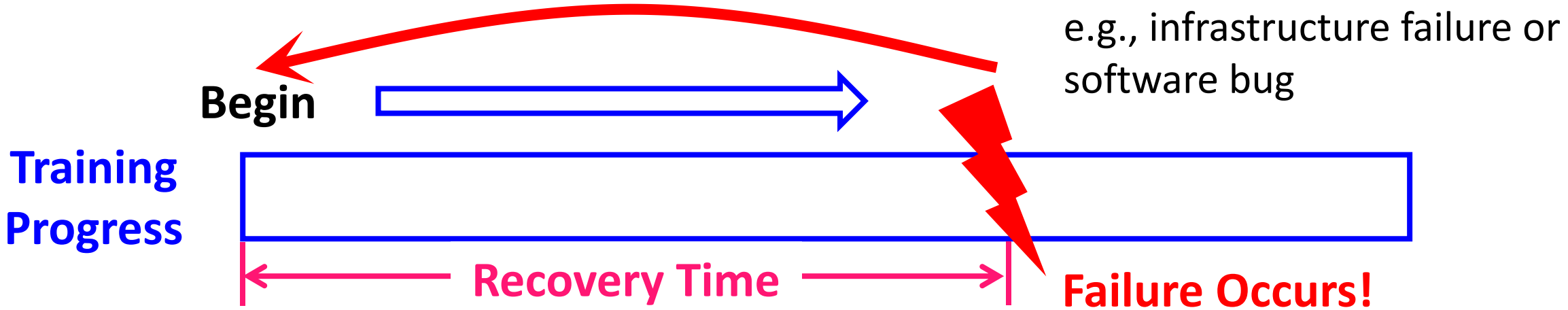


- **Checkpointing** is an efficient way to **ensure failure tolerance**

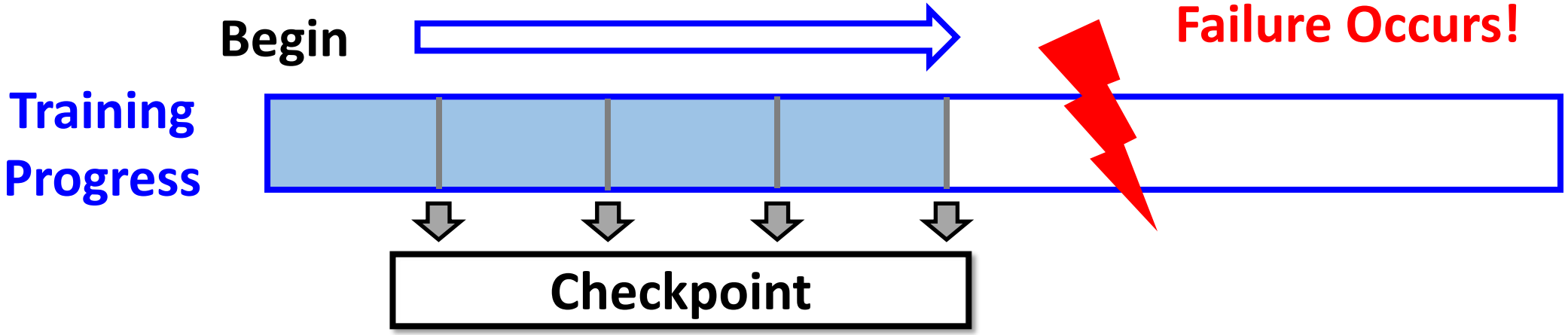


The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

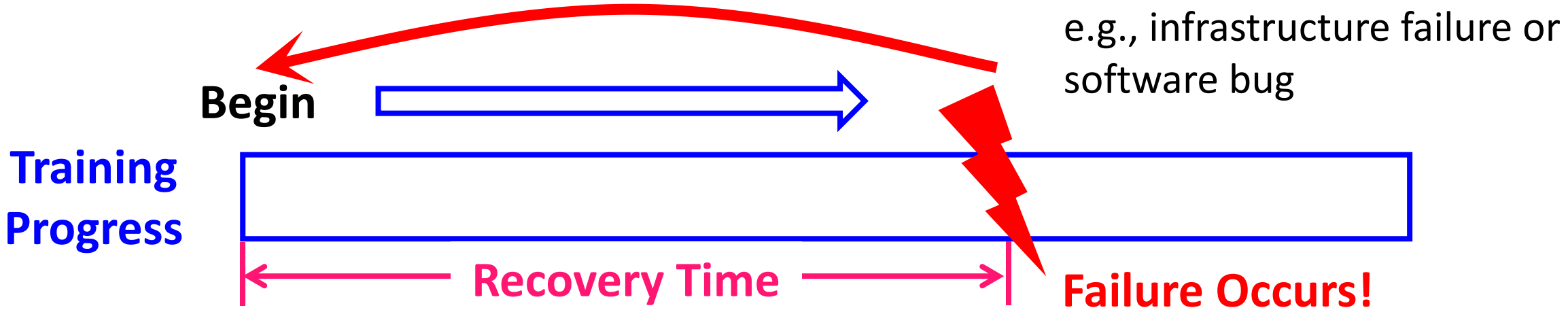


- **Checkpointing** is an efficient way to **ensure failure tolerance**

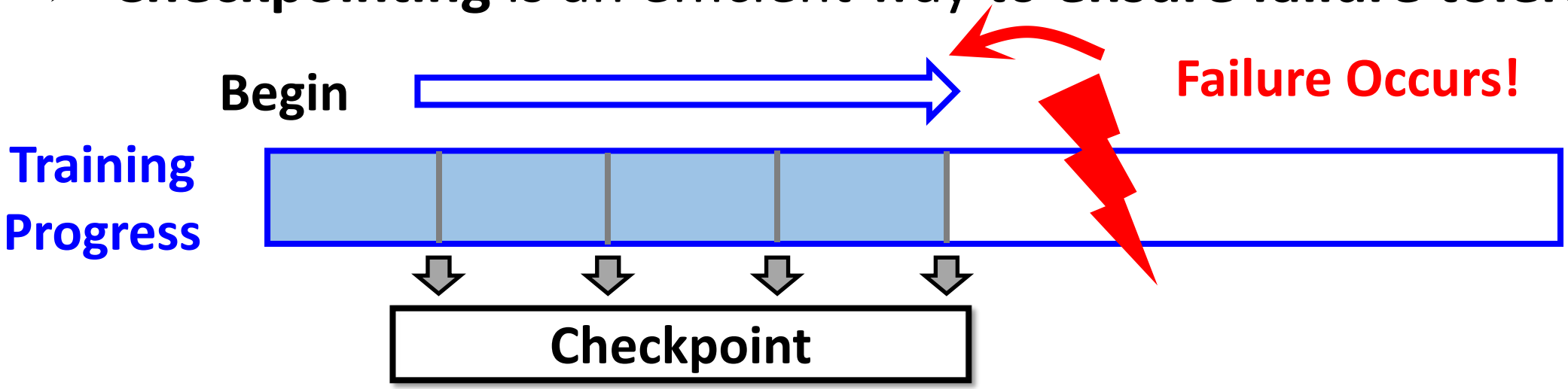


The importance of Failure Tolerance

- DNN training is **time-consuming** and **expensive**

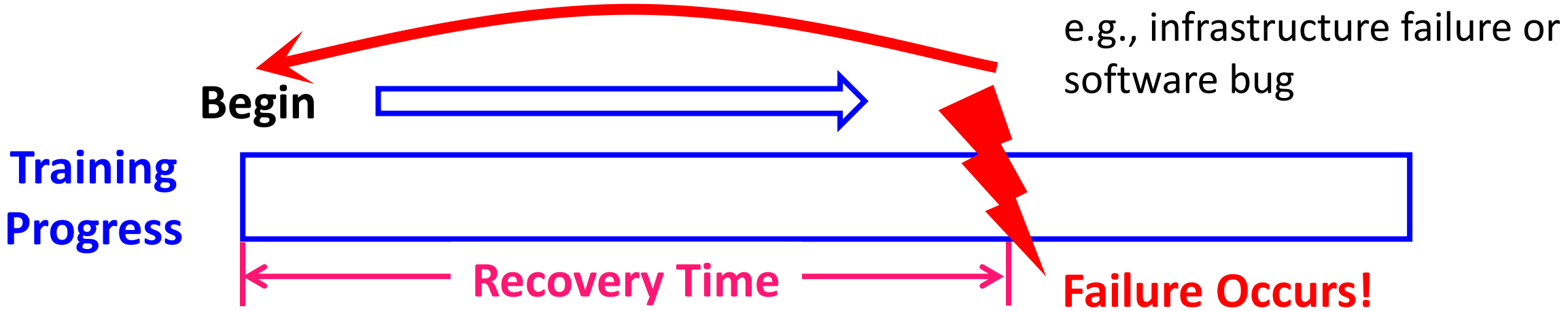


- **Checkpointing** is an efficient way to **ensure failure tolerance**

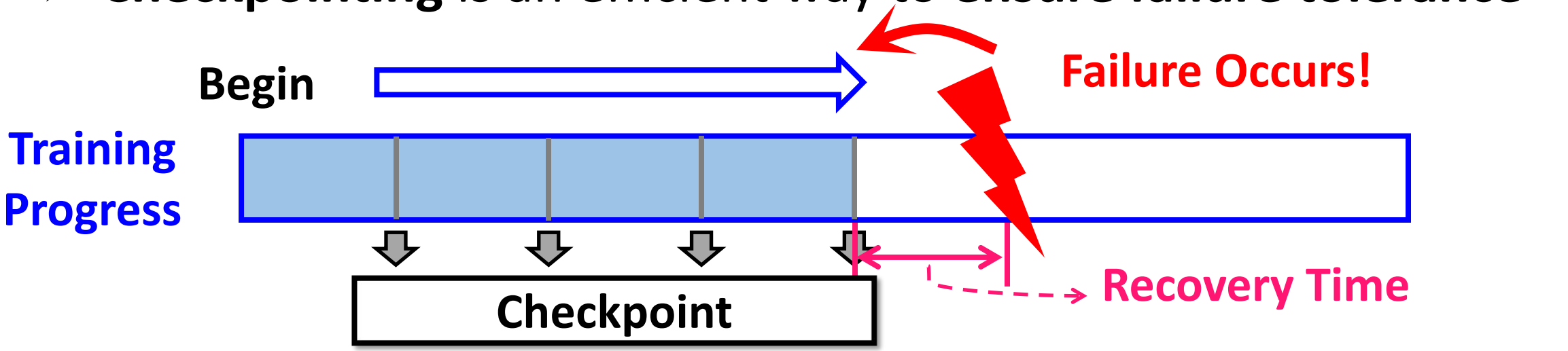


The importance of Failure Tolerance

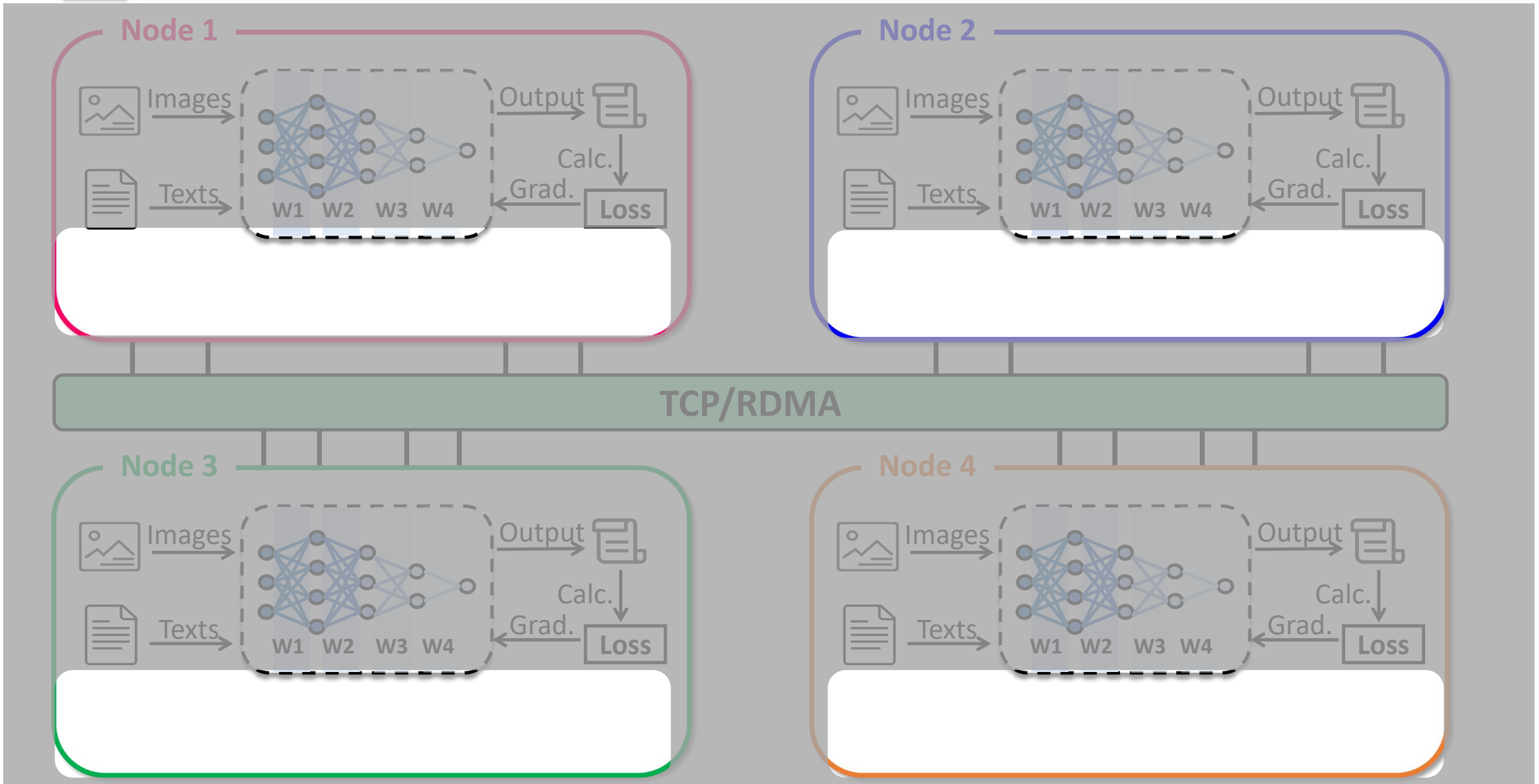
- DNN training is **time-consuming** and **expensive**



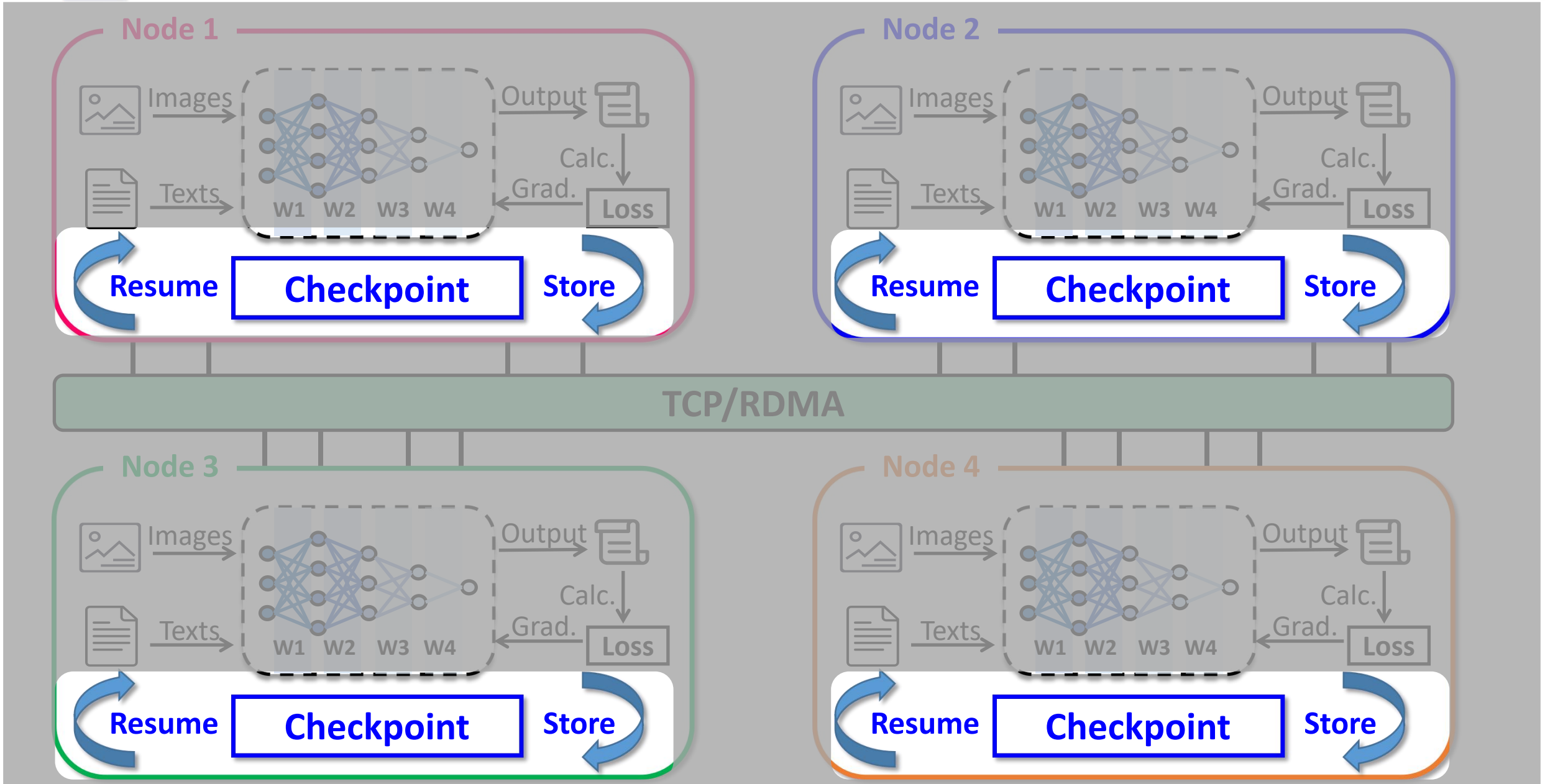
- **Checkpointing** is an efficient way to **ensure failure tolerance**



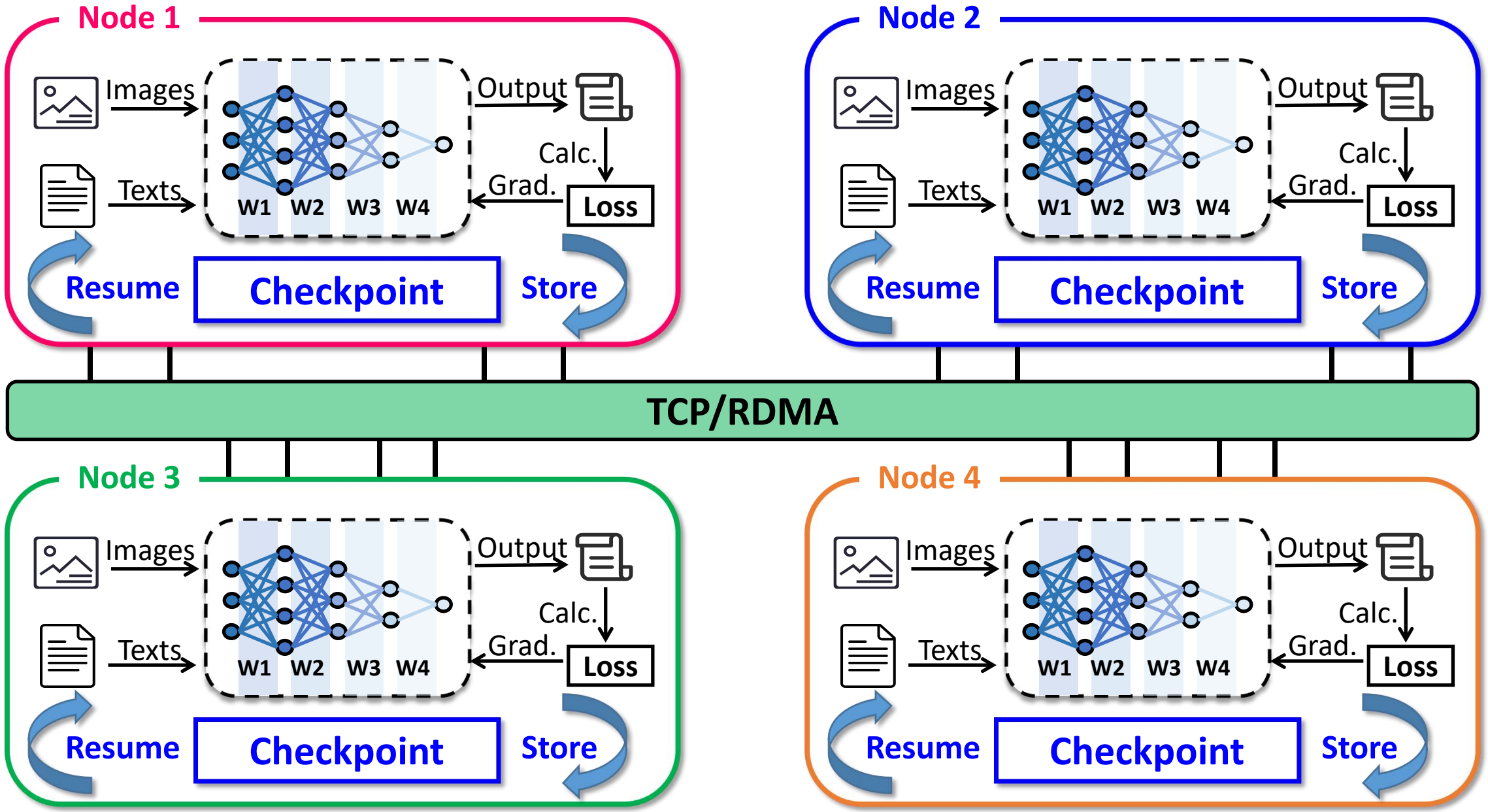
Checkpointing in Distributed DNN Training



Checkpointing in Distributed DNN Training



Checkpointing in Distributed DNN Training



The Need of Frequent Checkpointing

The Need of Frequent Checkpointing

- Failures are common in large-scale GPU clusters
  - **The mean time between failures** is low to **a few minutes**

The Need of Frequent Checkpointing

- Failures are common in large-scale GPU clusters
  - **The mean time between failures** is low to **a few minutes**
- Frequent job switches in the preemptive GPU cluster scheduling
  - **The interval between two switches** may be only **a few seconds**

The Need of Frequent Checkpointing

- Failures are common in large-scale GPU clusters
  - **The mean time between failures** is low to **a few minutes**
- Frequent job switches in the preemptive GPU cluster scheduling
  - **The interval between two switches** may be only **a few seconds**

Frequent Checkpointing

The Need of Frequent Checkpointing

- Failures are common in large-scale GPU clusters
  - **The mean time between failures** is low to **a few minutes**
- Frequent job switches in the preemptive GPU cluster scheduling
  - **The interval between two switches** may be only **a few seconds**

Frequent Checkpointing

⇩

High Runtime Overhead

The Need of Frequent Checkpointing

- Failures are common in large-scale GPU clusters
  - **The mean time between failures** is low to **a few minutes**
- Frequent job switches in the preemptive GPU cluster scheduling
  - **The interval between two switches** may be only **a few seconds**

Frequent Checkpointing

⇩

High Runtime Overhead

Existing Checkpointing Schemes are Inefficient

[1] PyTorch@NIPS'19 [2] SCAR@ICML'19 [3] DeepFreeze@CCGRID'20 [4] CheckFreq@FAST'21

Existing Checkpointing Schemes are Inefficient

- **Synchronous checkpointing**[1]
  - Introduce **severe training stall**
  - Suffer from **high runtime overhead**

[1] PyTorch@NIPS'19 [2] SCAR@ICML'19 [3] DeepFreeze@CCGRID'20 [4] CheckFreq@FAST'21

Existing Checkpointing Schemes are Inefficient

- **Synchronous checkpointing**[1]
  - Introduce **severe training stall**
  - Suffer from **high runtime overhead**
- **Asynchronous checkpointing**[2-4]
  - Two-phase checkpointing
  - Pipeline the checkpointing with computation

[1] PyTorch@NIPS'19 [2] SCAR@ICML'19 [3] DeepFreeze@CCGRID'20 [4] CheckFreq@FAST'21

Existing Checkpointing Schemes are Inefficient

- **Synchronous checkpointing**[1]
  - Introduce **severe training stall**
  - Suffer from **high runtime overhead**
- **Asynchronous checkpointing**[2-4]
  - Two-phase checkpointing
  - Pipeline the checkpointing with computation
  - Sub-optimal due to **monolithic** checkpointing process
  - **Fail to** fully pipeline checkpointing with communication

[1] PyTorch@NIPS'19 [2] SCAR@ICML'19 [3] DeepFreeze@CCGRID'20 [4] CheckFreq@FAST'21

Persistent Memory (PM)

- Intel Optane PM
- Samsung Memory-Semantic CXL (Compute Express Link) SSD

Persistent Memory (PM)

- Intel Optane PM
- Samsung Memory-Semantic CXL (Compute Express Link) SSD

OR

Persistent Memory (PM)

- Intel Optane PM
- Samsung Memory-Semantic CXL (Compute Express Link) SSD

OR

Byte-addressable **Fine-grained Persistence** **Near-DRAM performance**

Our Design

Our Design

LightCheck: A cost-efficient checkpointing scheme for distribued DNN training

Our Design

LightCheck: A cost-efficient checkpointing scheme for distribued DNN training

- **Asynchronous layer-wise checkpointing**
  - Fine-grained pipelining
  - Communication-aware

Our Design

LightCheck: A cost-efficient checkpointing scheme for distribued DNN training

- **Asynchronous layer-wise checkpointing**
  - Fine-grained pipelining
  - Communication-aware
- **Efficient persistent memory management**
  - Direct access
  - Metadata-aware

Our Design

LightCheck: A cost-efficient checkpointing scheme for distribued DNN training

- **Asynchronous layer-wise checkpointing**
  - Fine-grained pipelining
  - Communication-aware
  - ➡ **Minimizing training stalls**
- **Efficient persistent memory management**
  - Direct access
  - Metadata-aware

Our Design

LightCheck: A cost-efficient checkpointing scheme for distribued DNN training

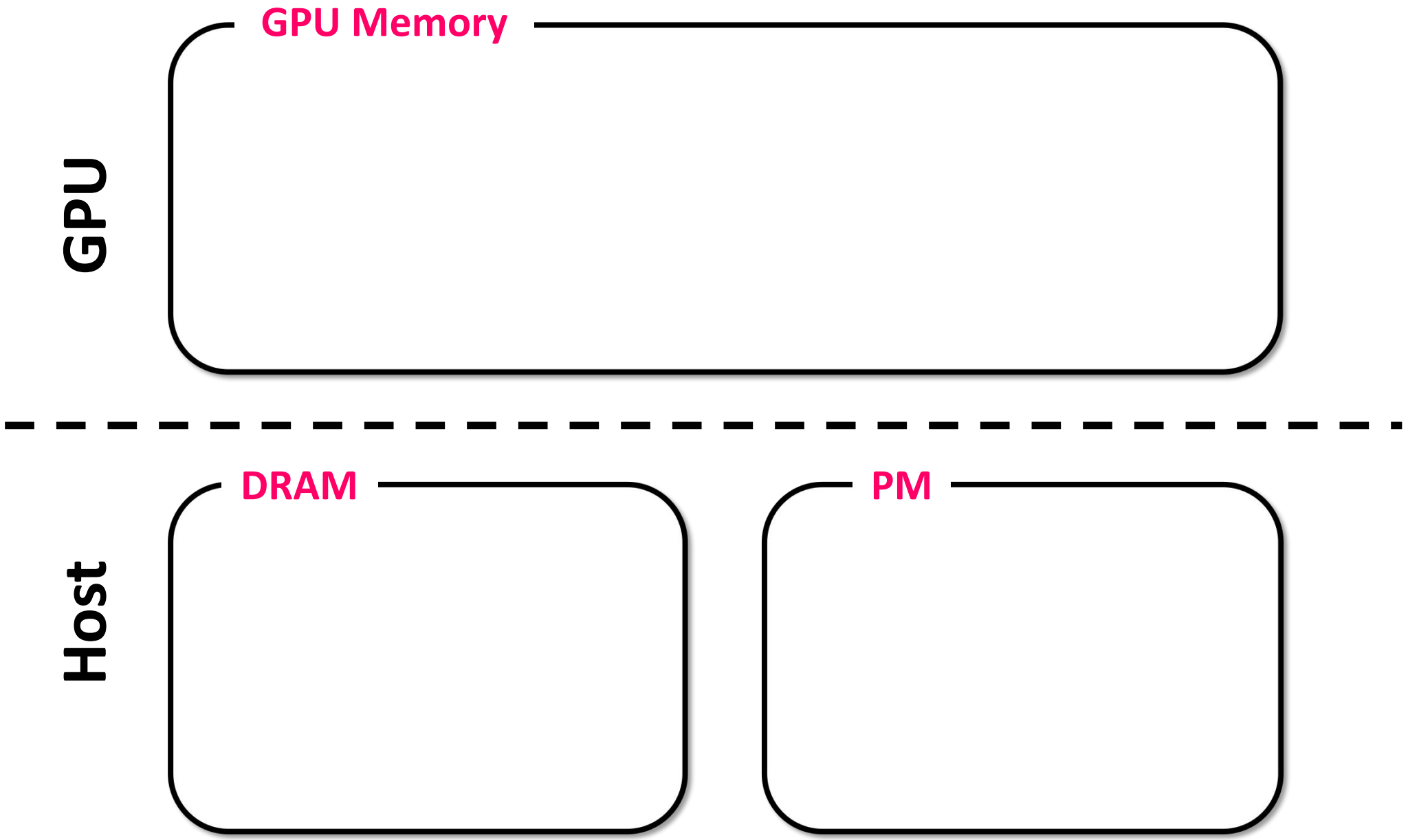
- **Asynchronous layer-wise checkpointing**
  - Fine-grained pipelining
  - Communication-aware

  ➡ **Minimizing training stalls**

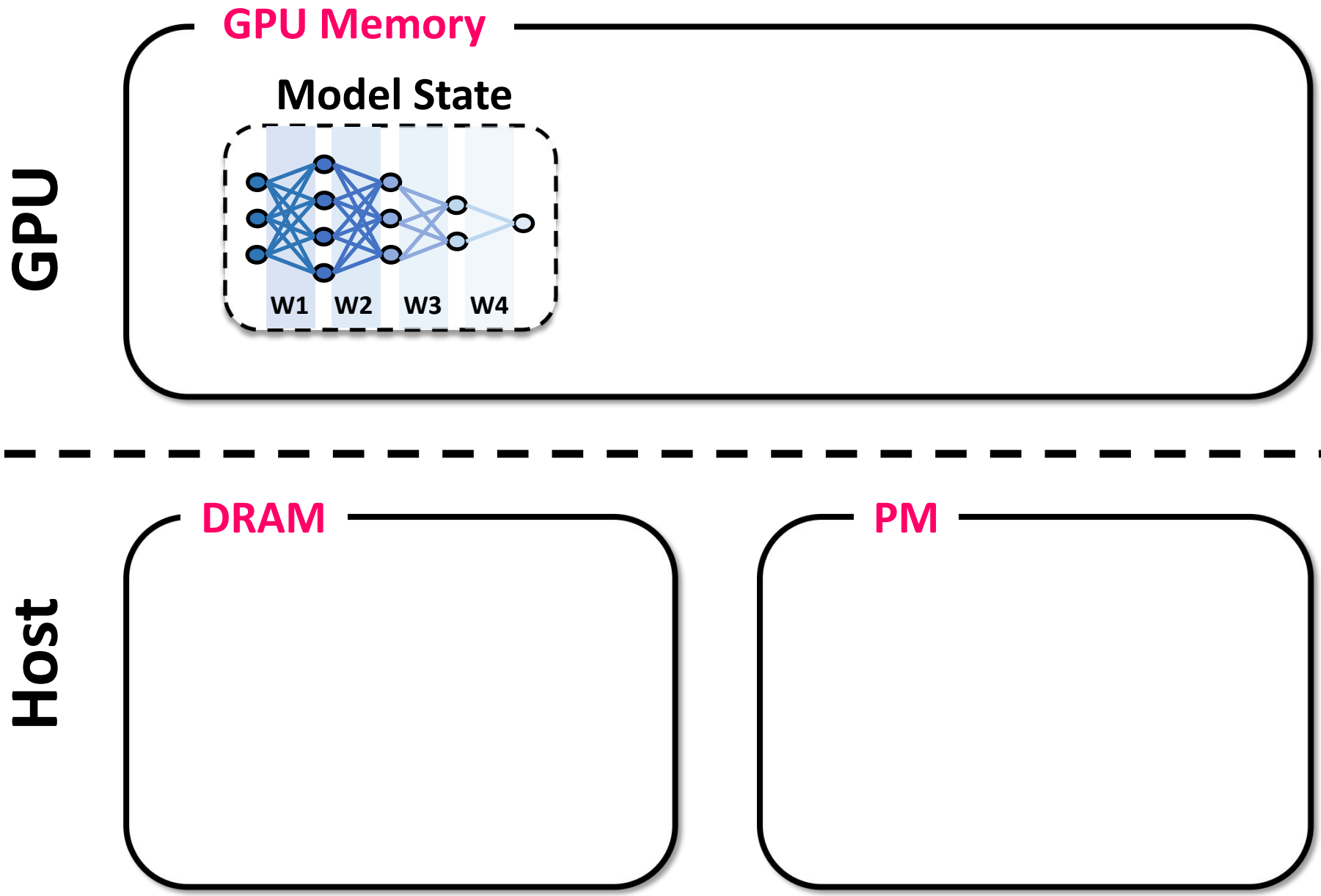
- **Efficient persistent memory management**
  - Direct access
  - Metadata-aware

  ➡ **Fully exploiting persistent memory**

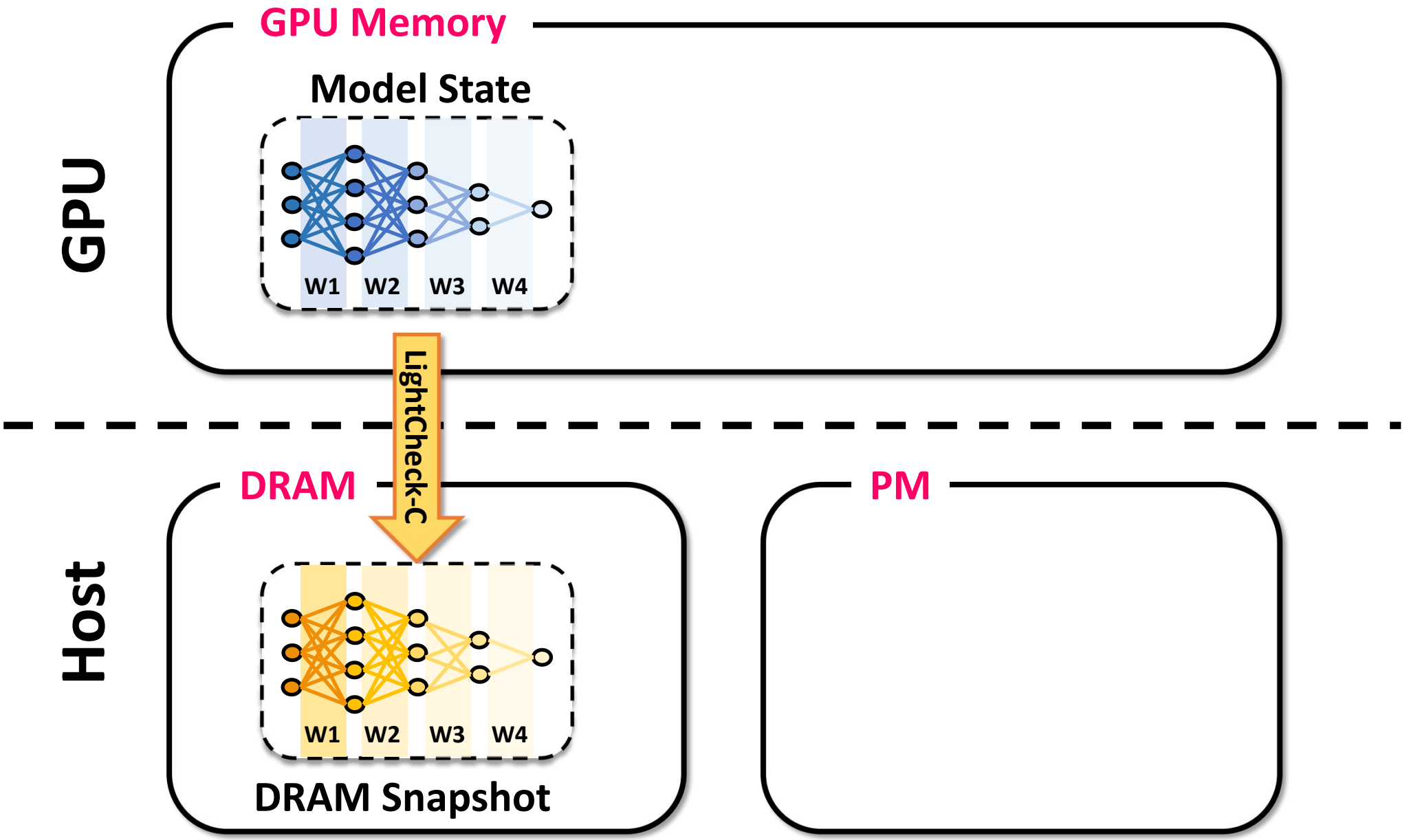
Checkpointing Strategies



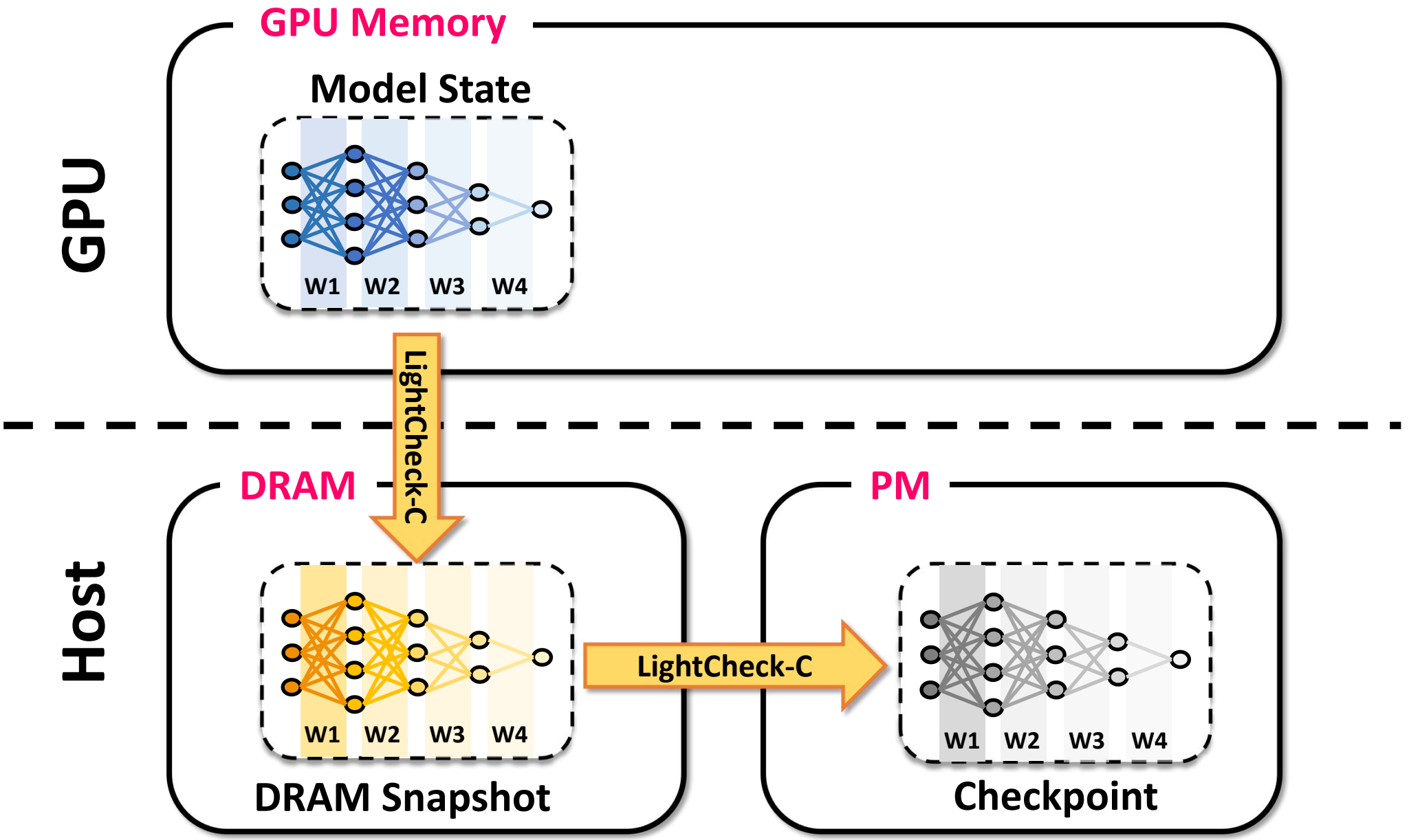
Checkpointing Strategies



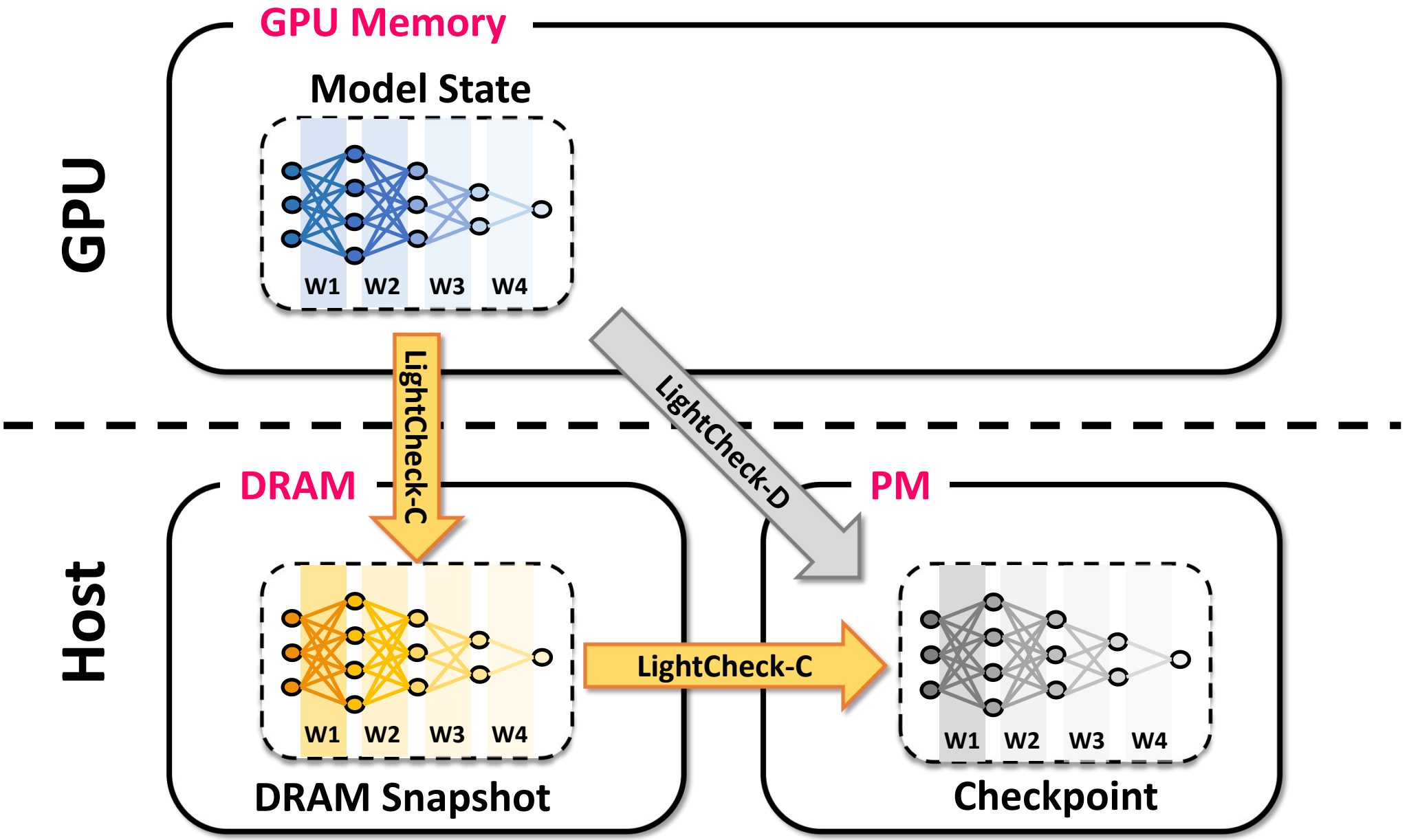
Checkpointing Strategies



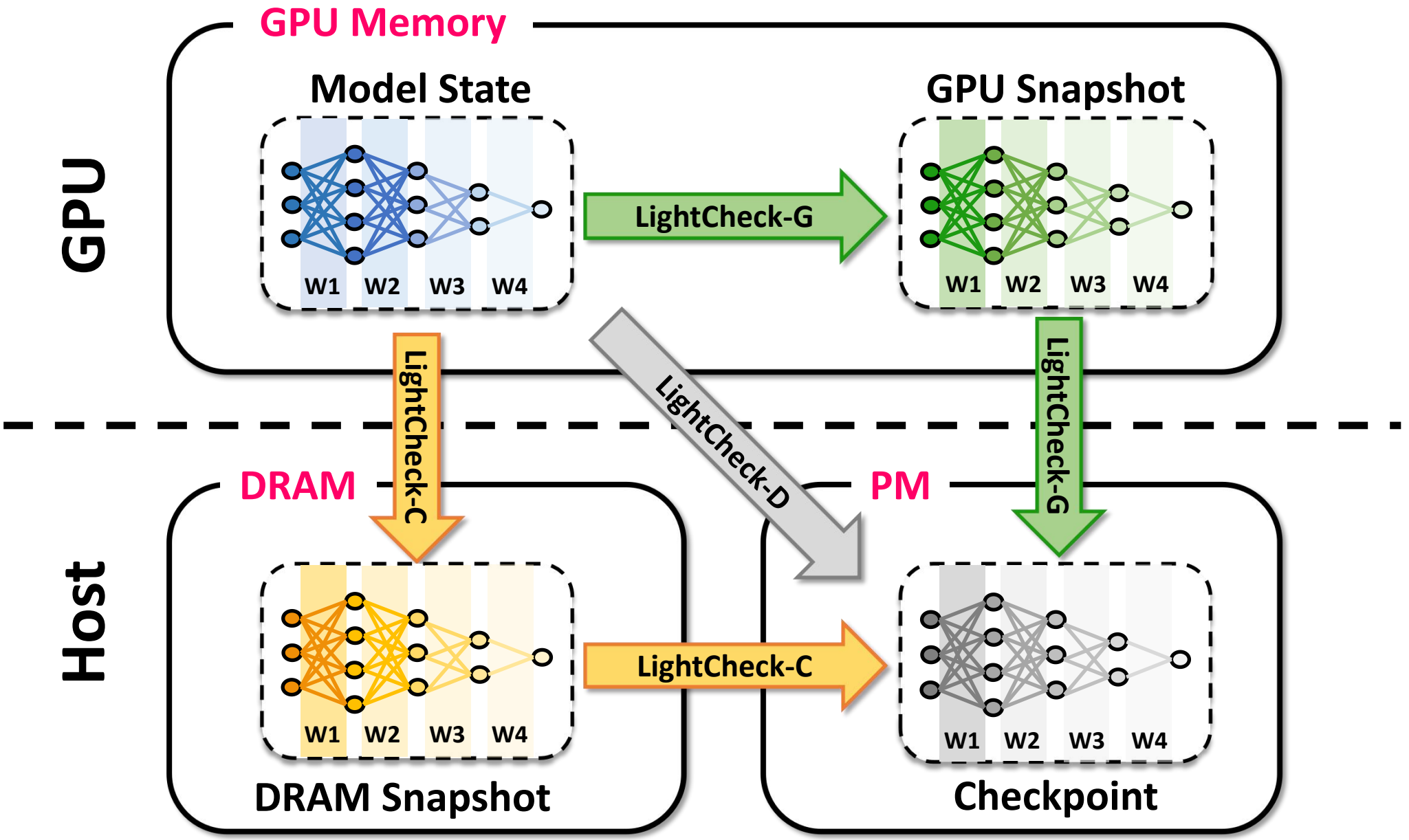
Checkpointing Strategies



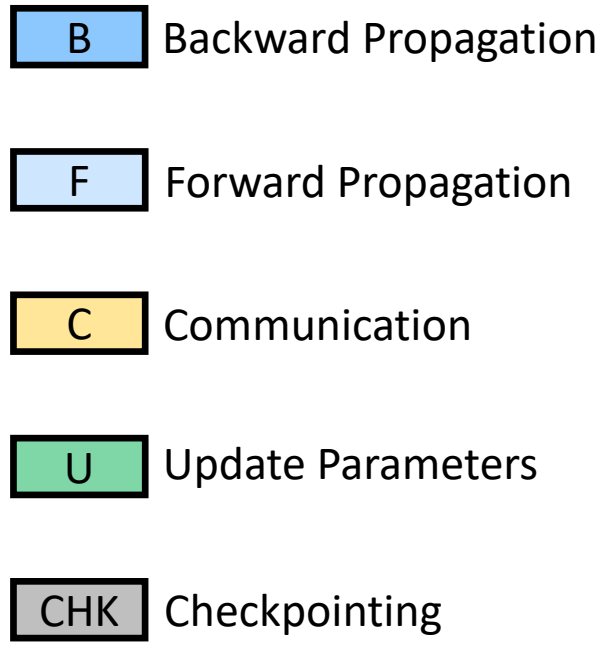
Checkpointing Strategies



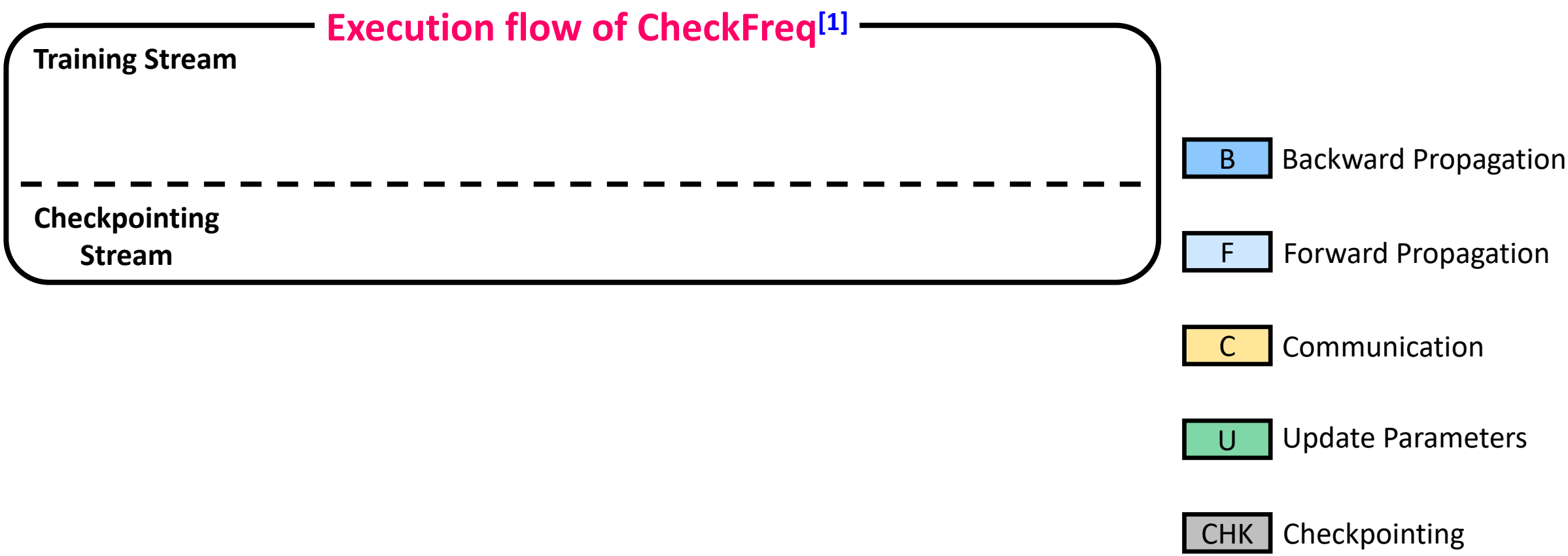
Checkpointing Strategies



Asynchronous Layer-wise Checkpointing

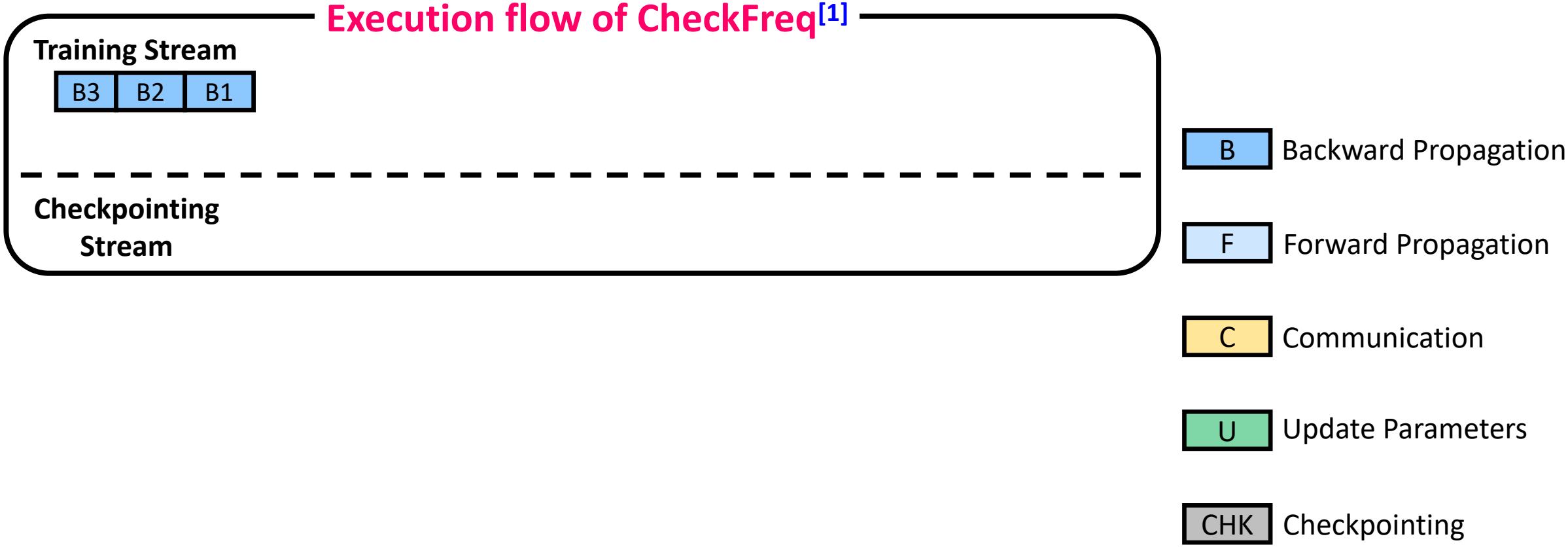


Asynchronous Layer-wise Checkpointing



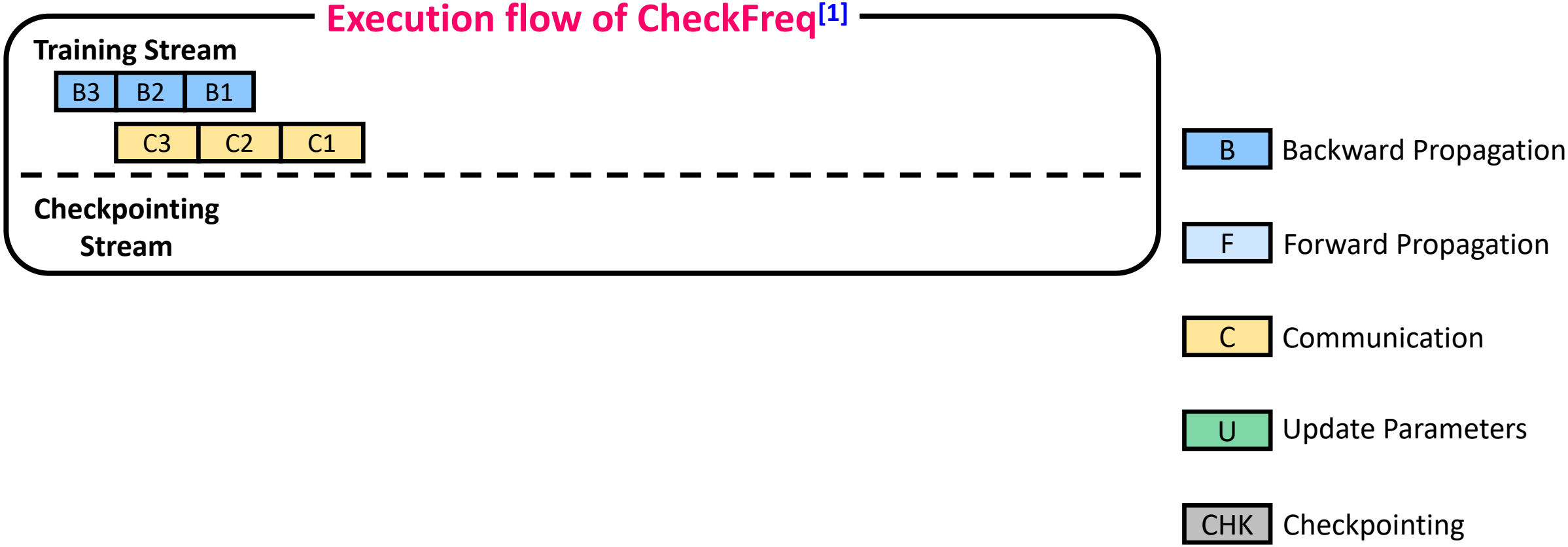
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



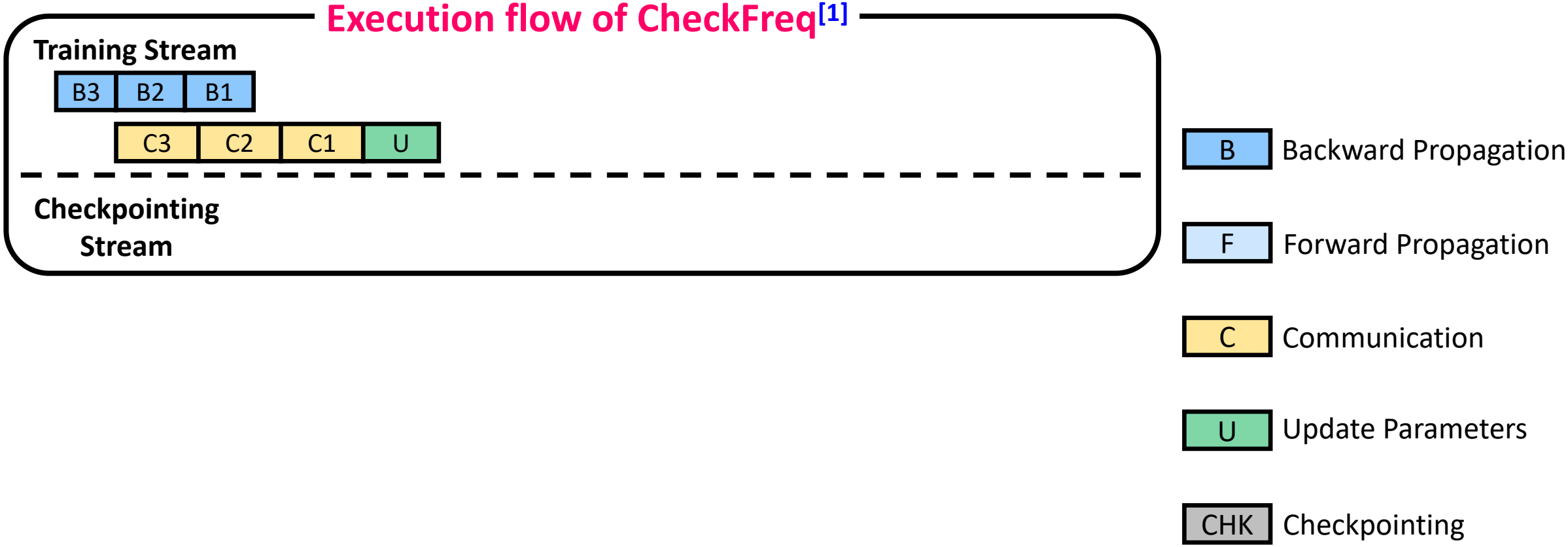
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



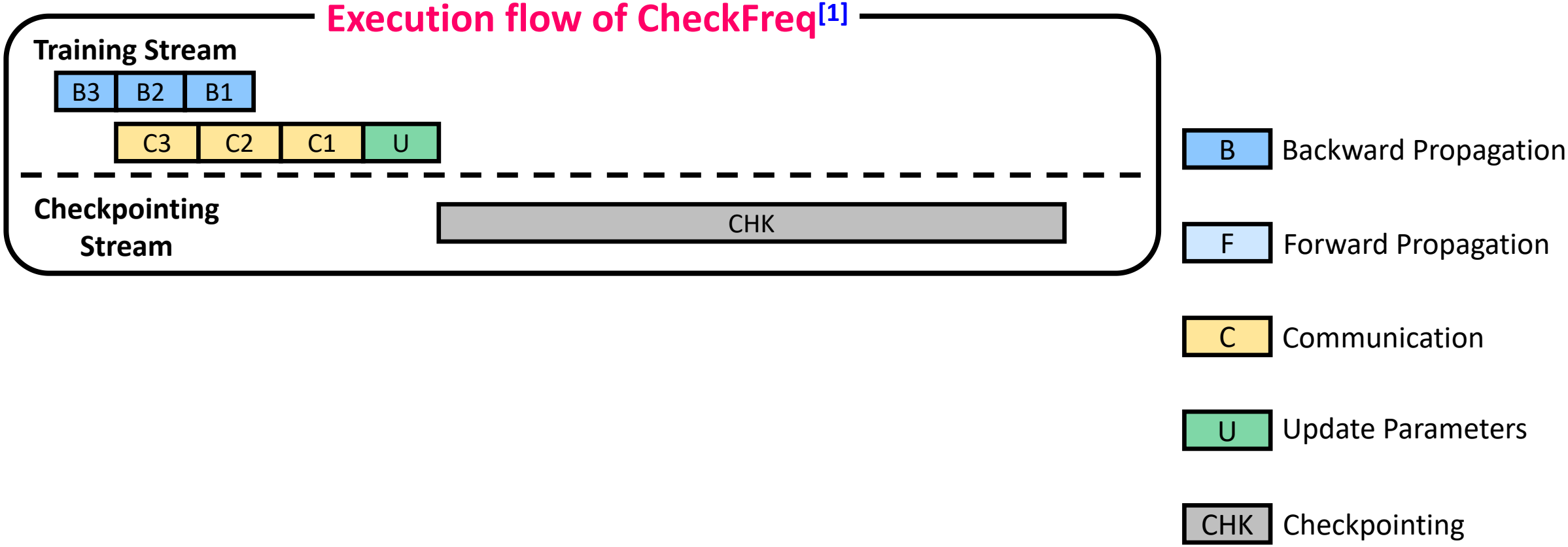
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



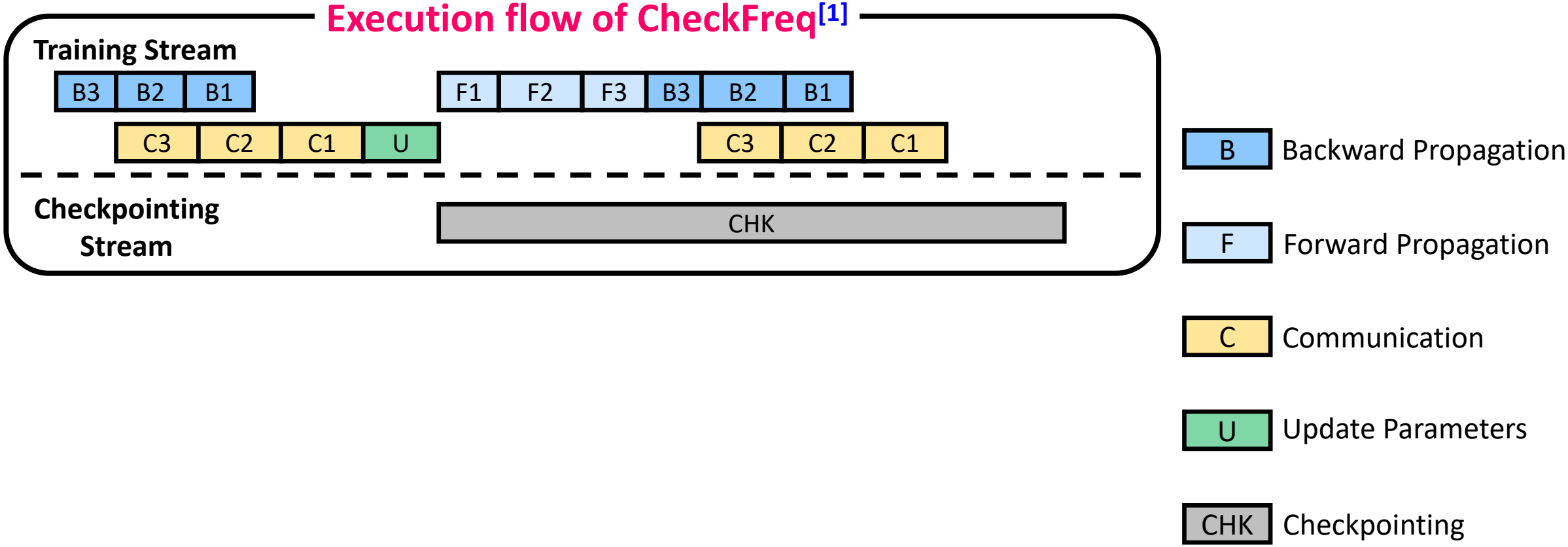
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



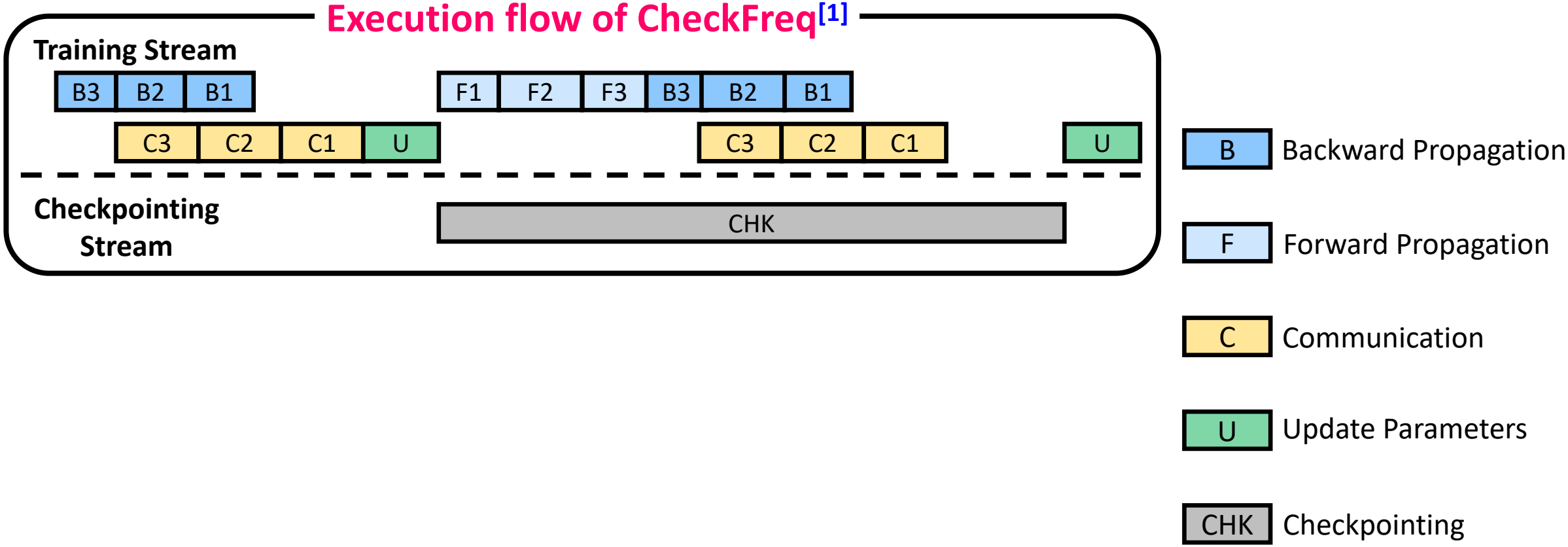
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



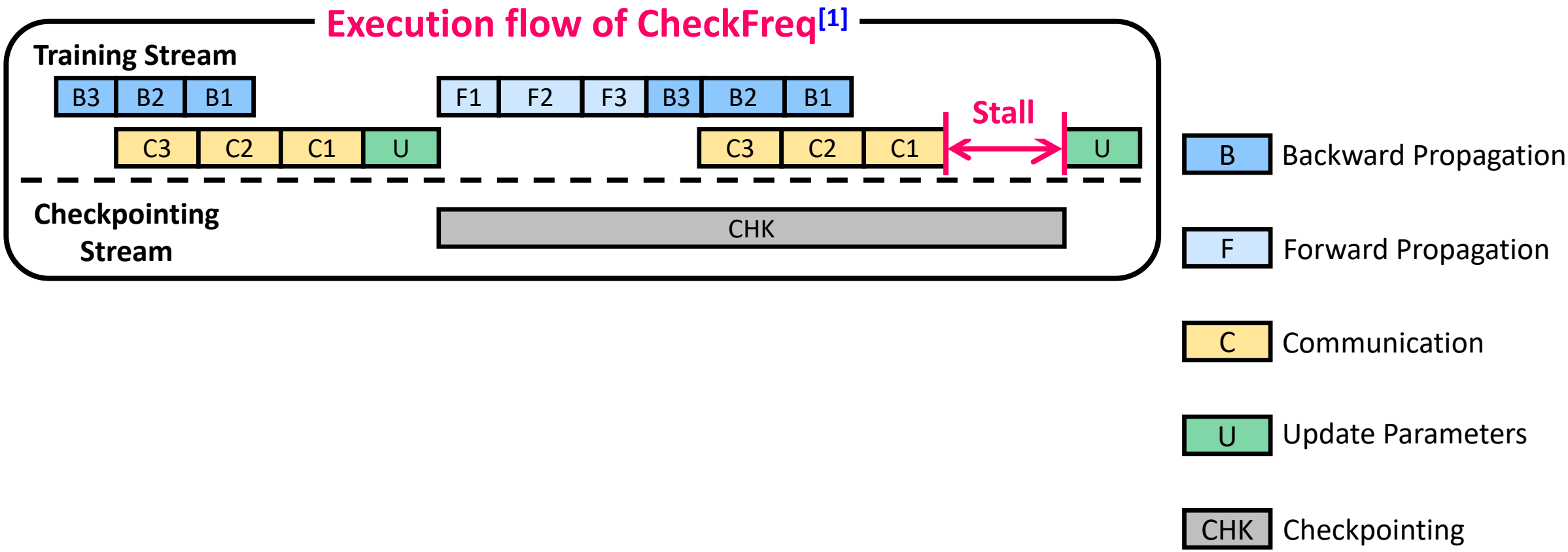
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



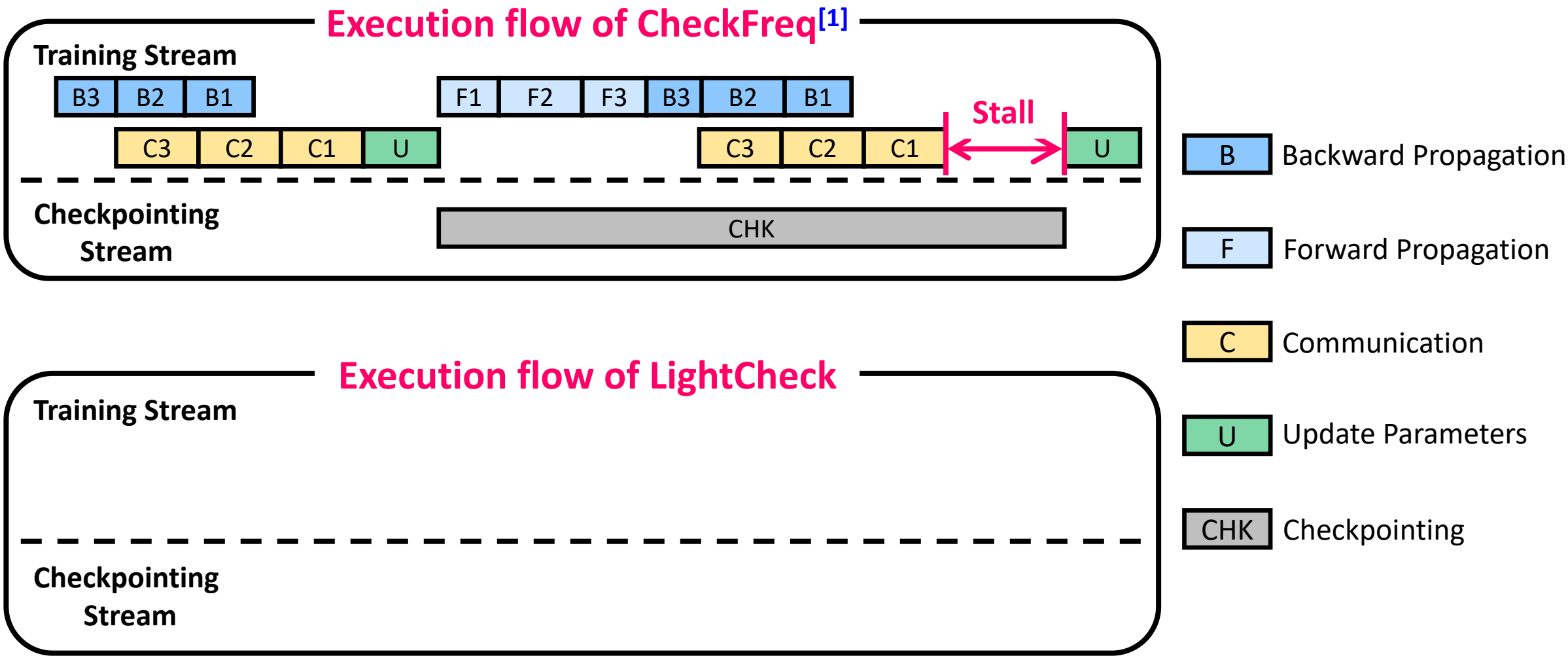
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



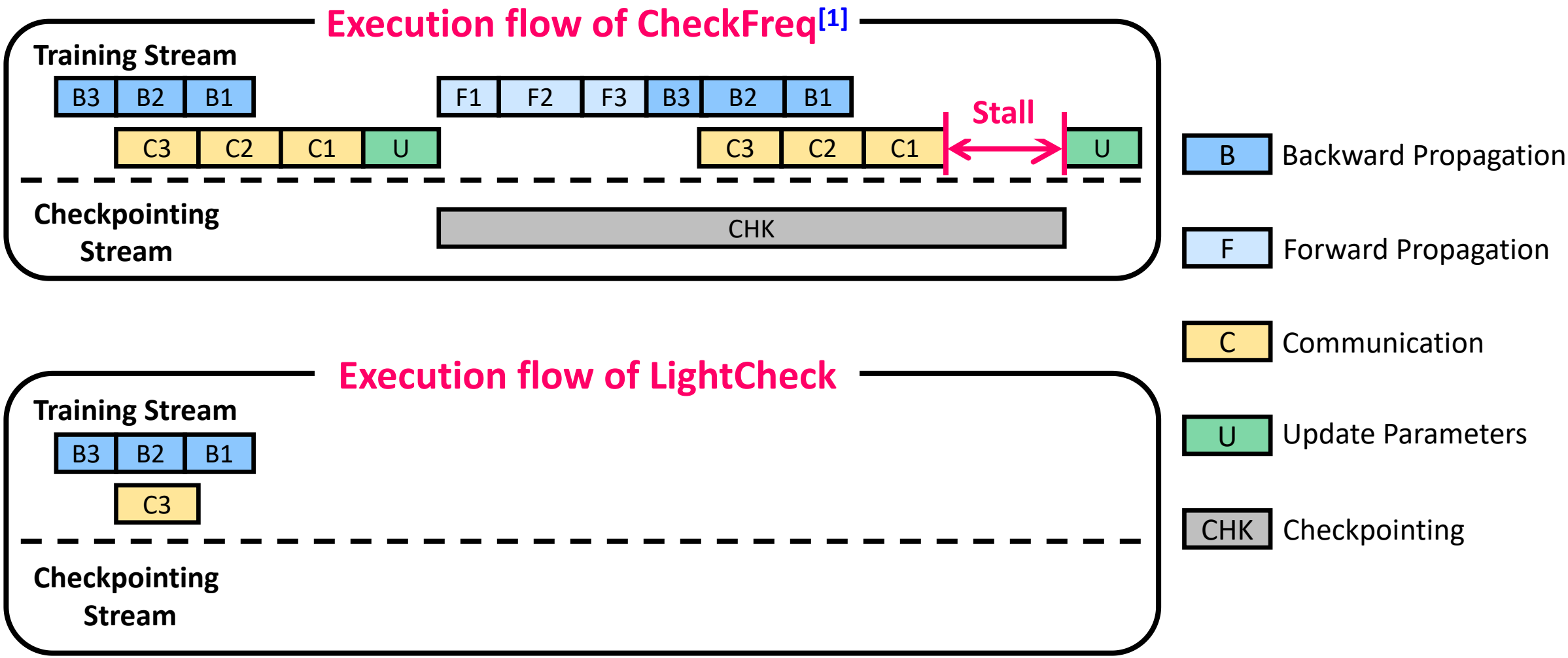
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



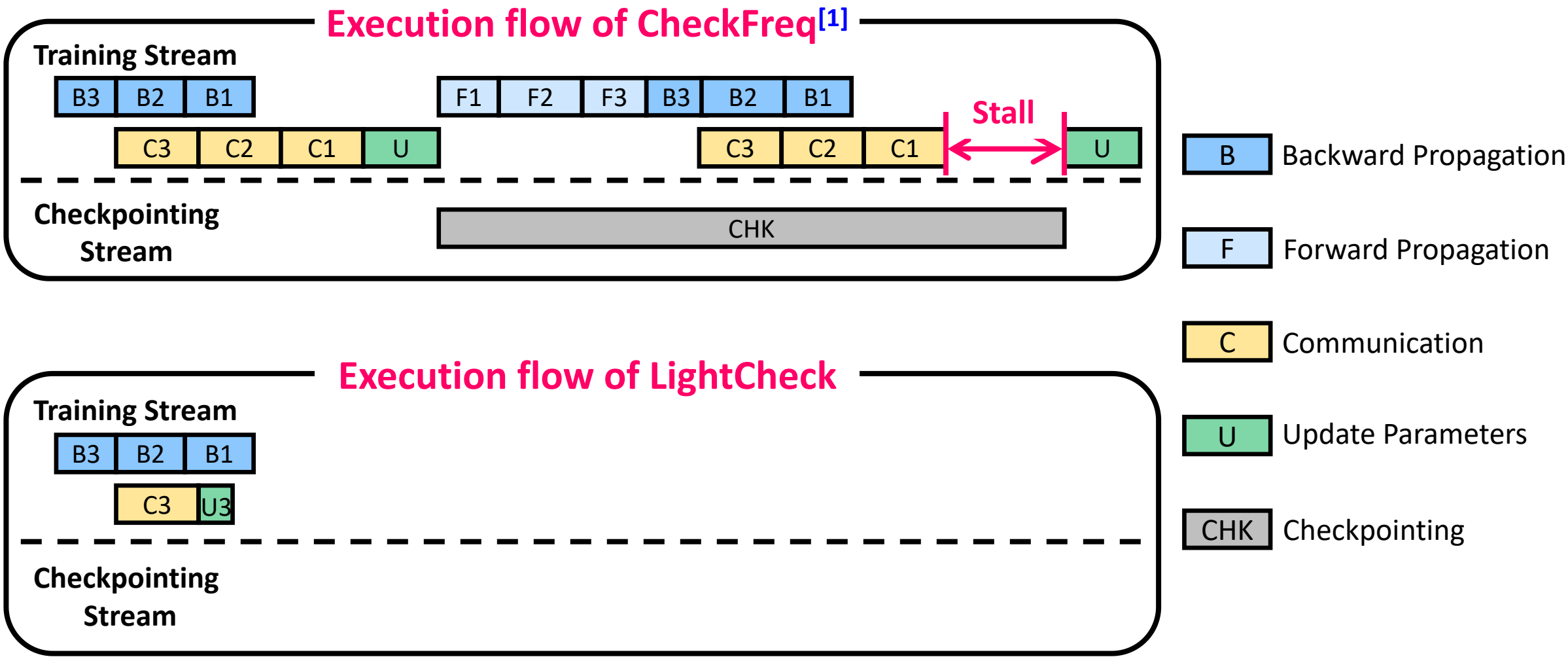
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



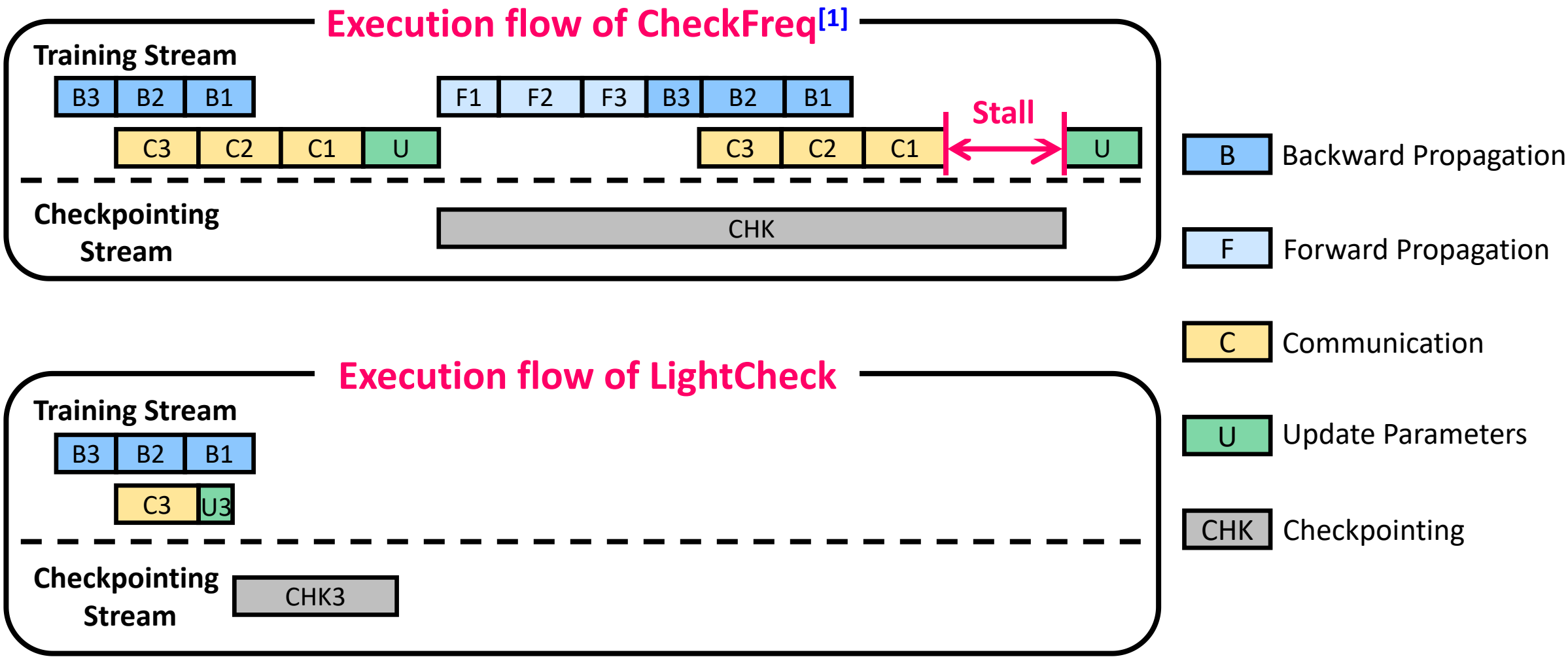
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



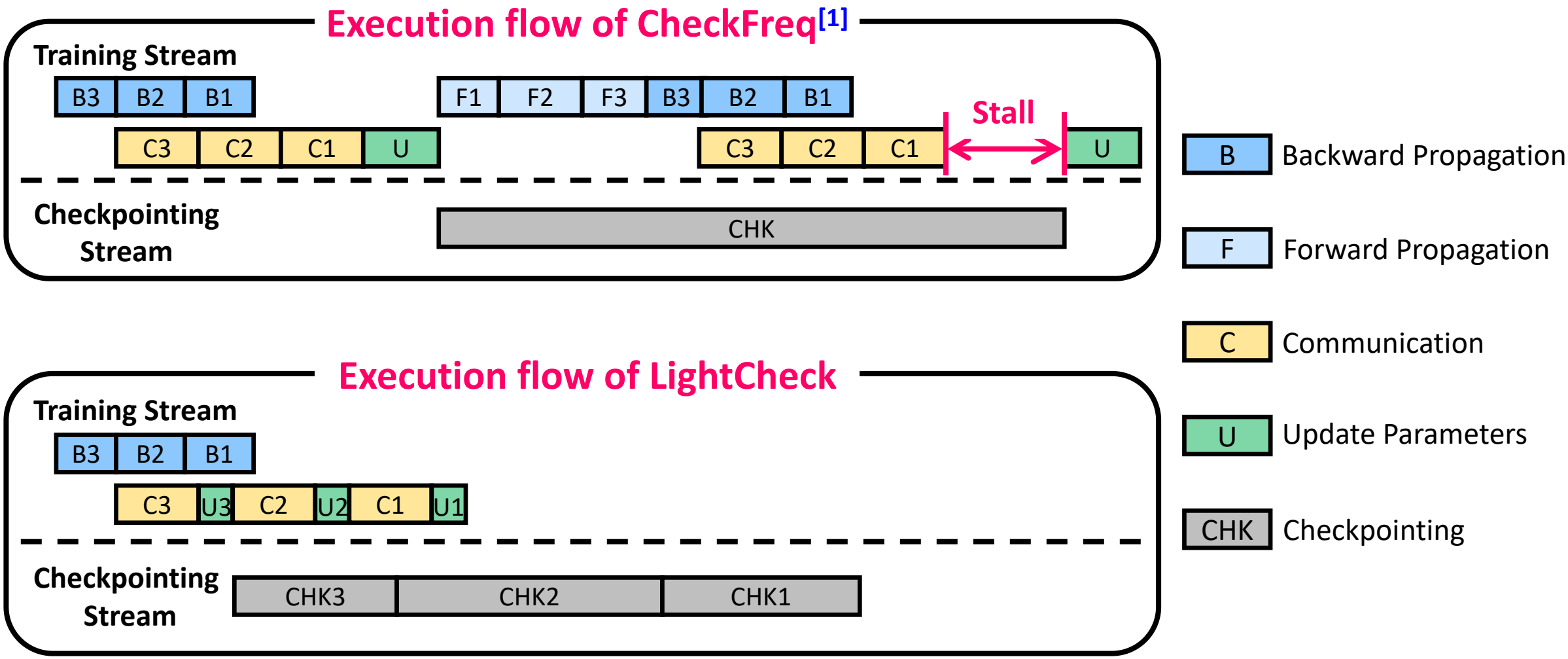
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



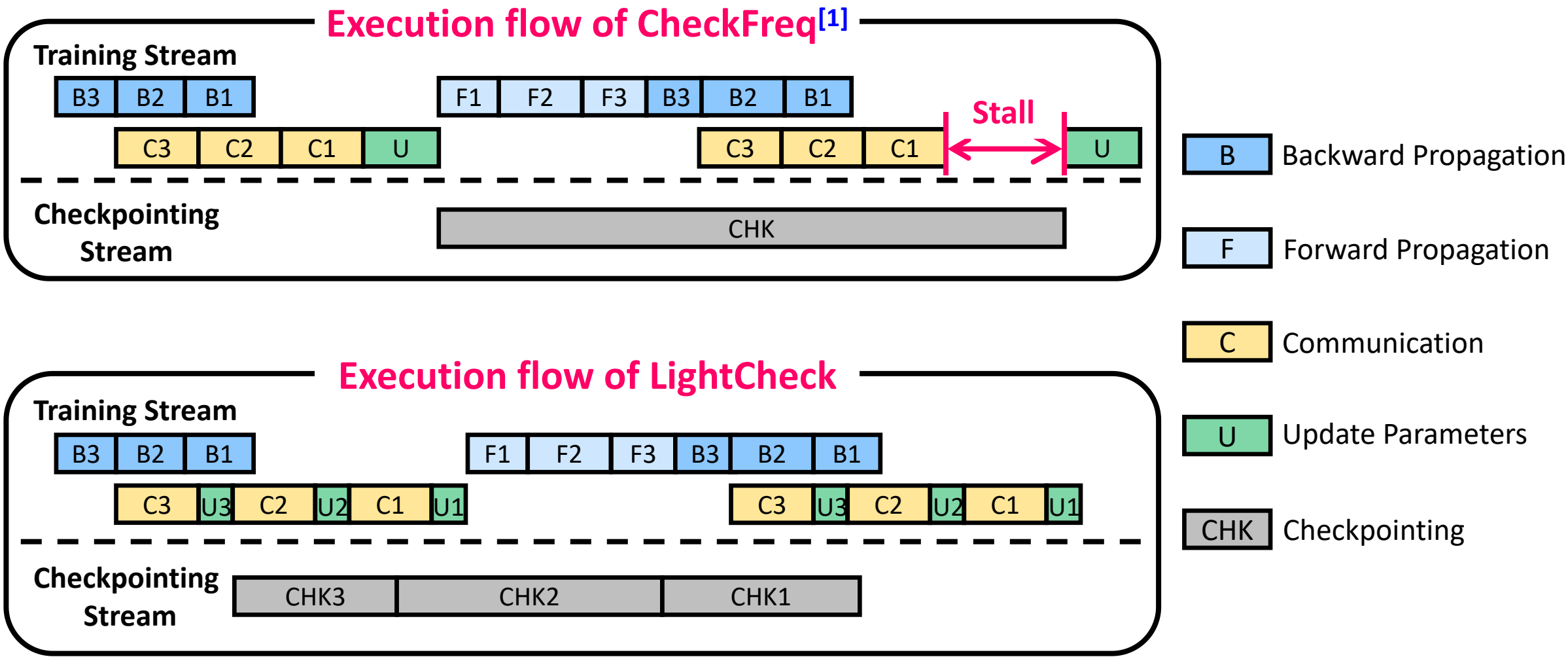
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



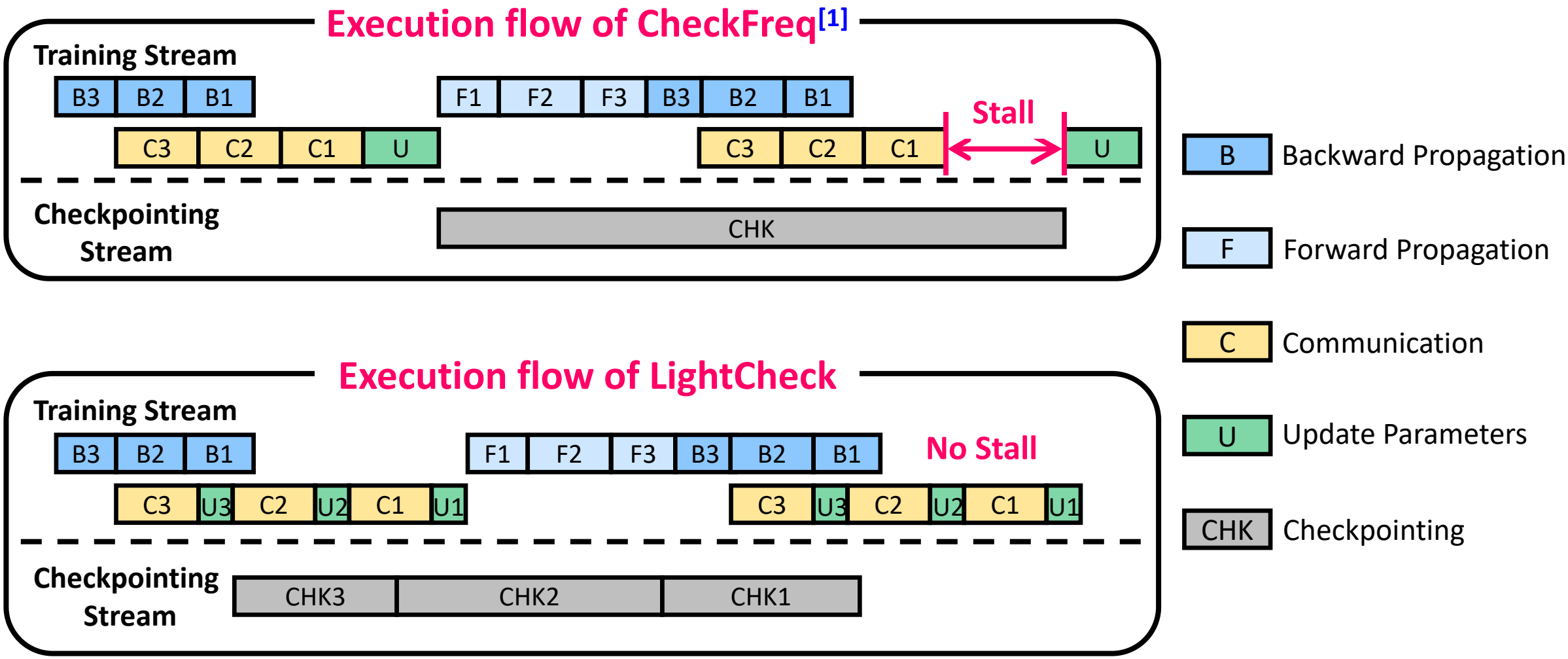
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

Asynchronous Layer-wise Checkpointing



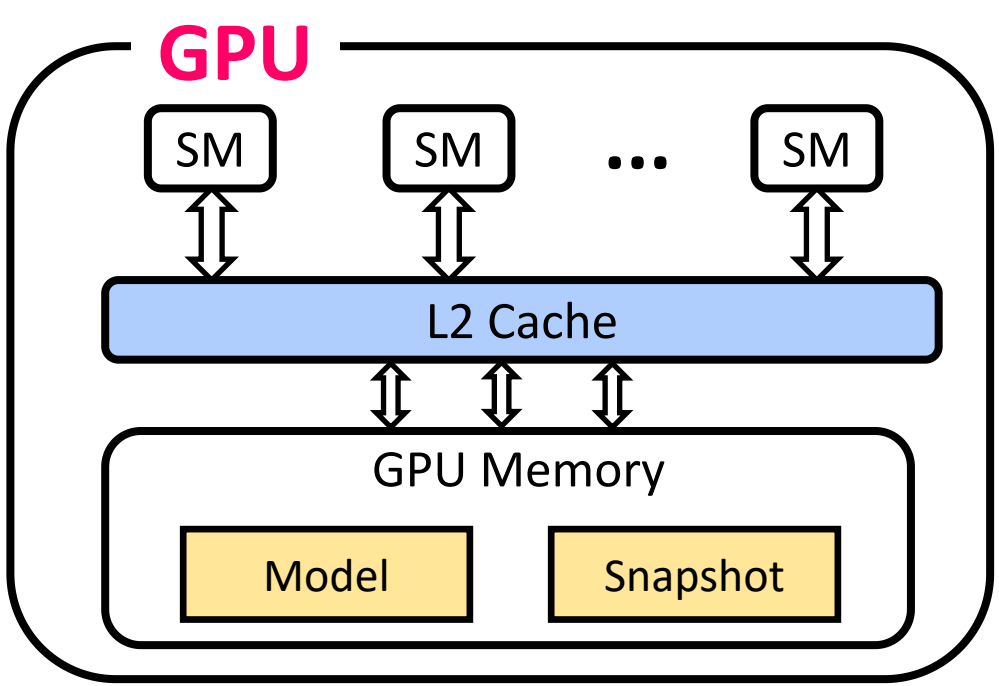
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

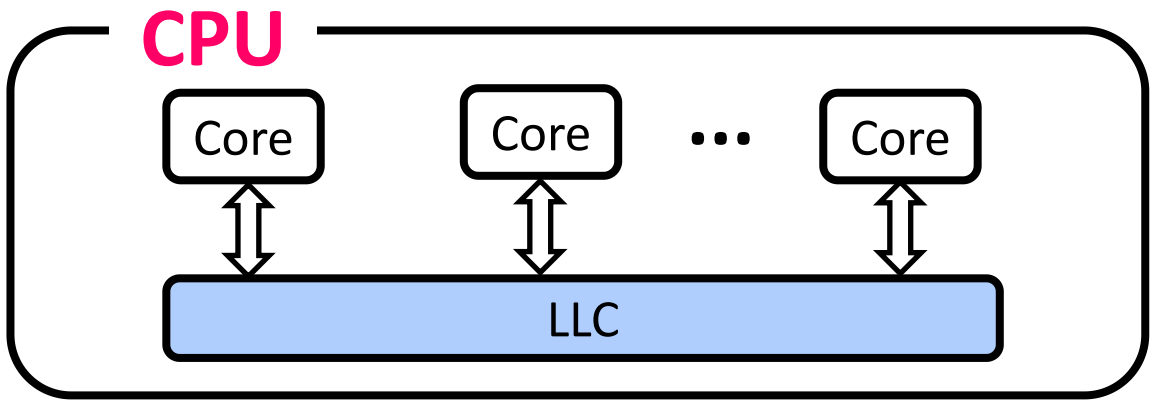
Asynchronous Layer-wise Checkpointing



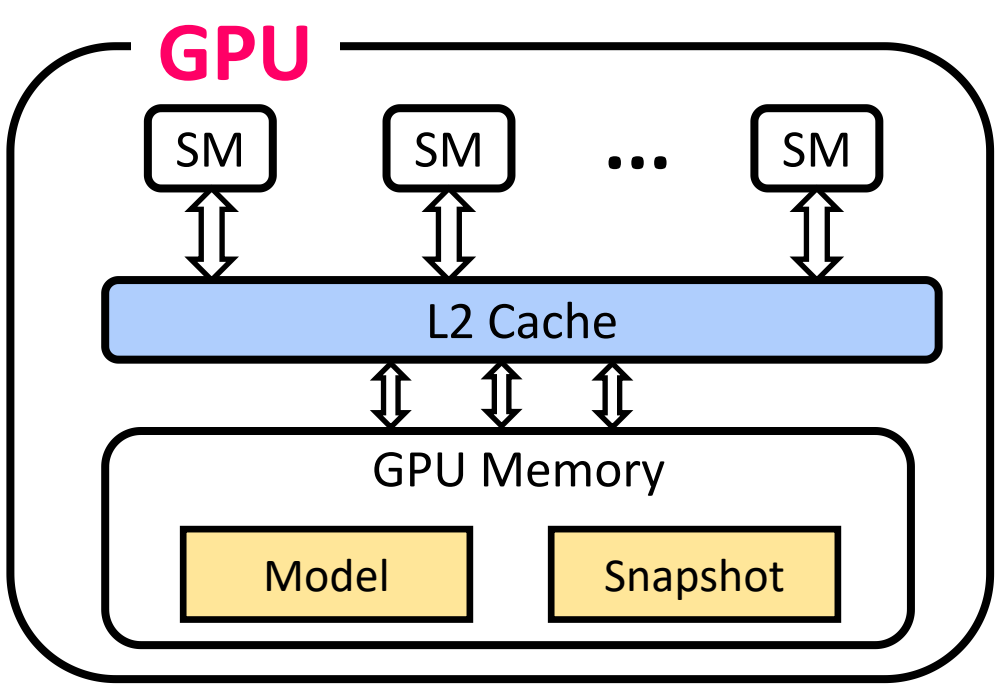
[1] J. Mohan, A. Phanishayee, and V. Chidambaram, “Checkfreq: Frequent, fine-grained dnn checkpointing,” in FAST, 2021

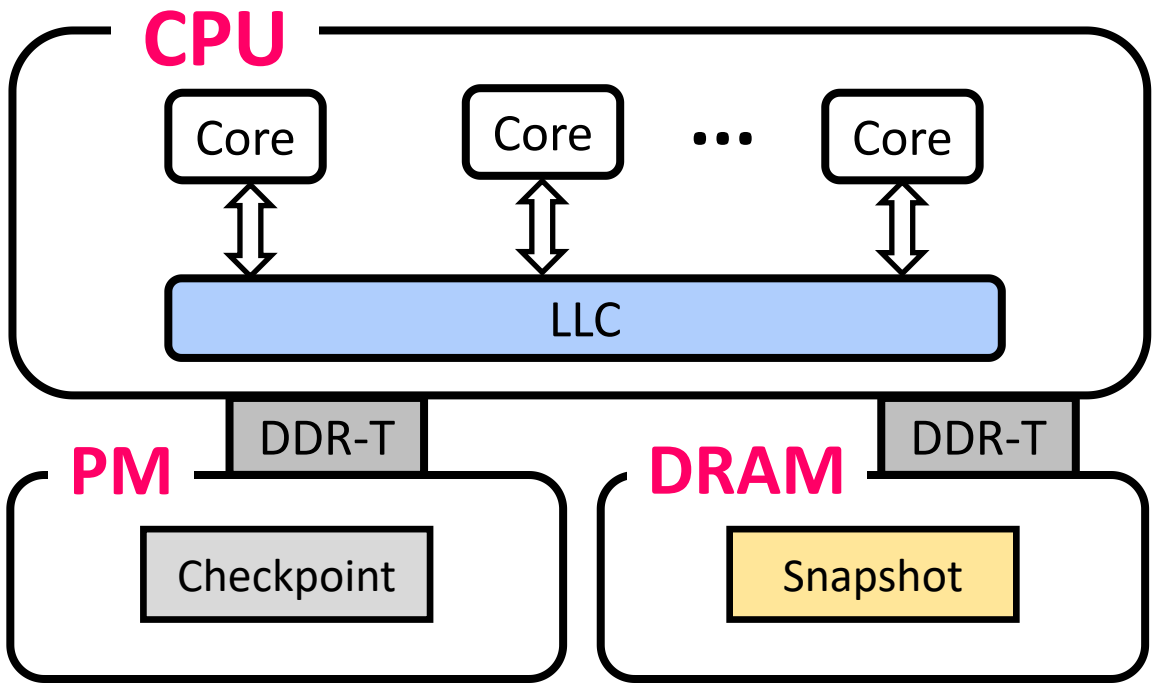
Efficient persistent memory management



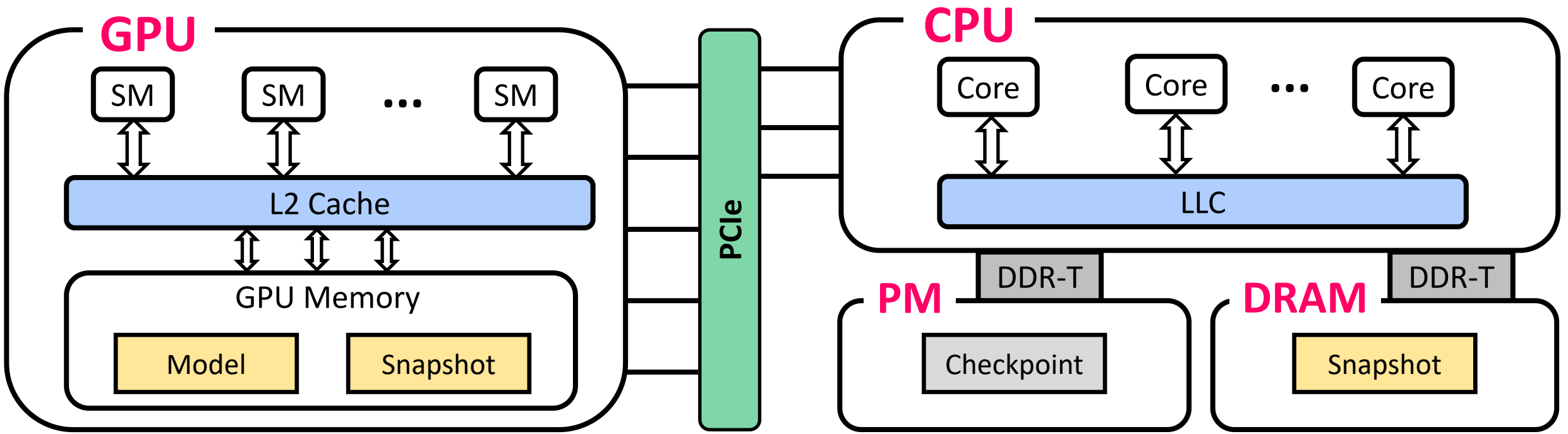


Efficient persistent memory management

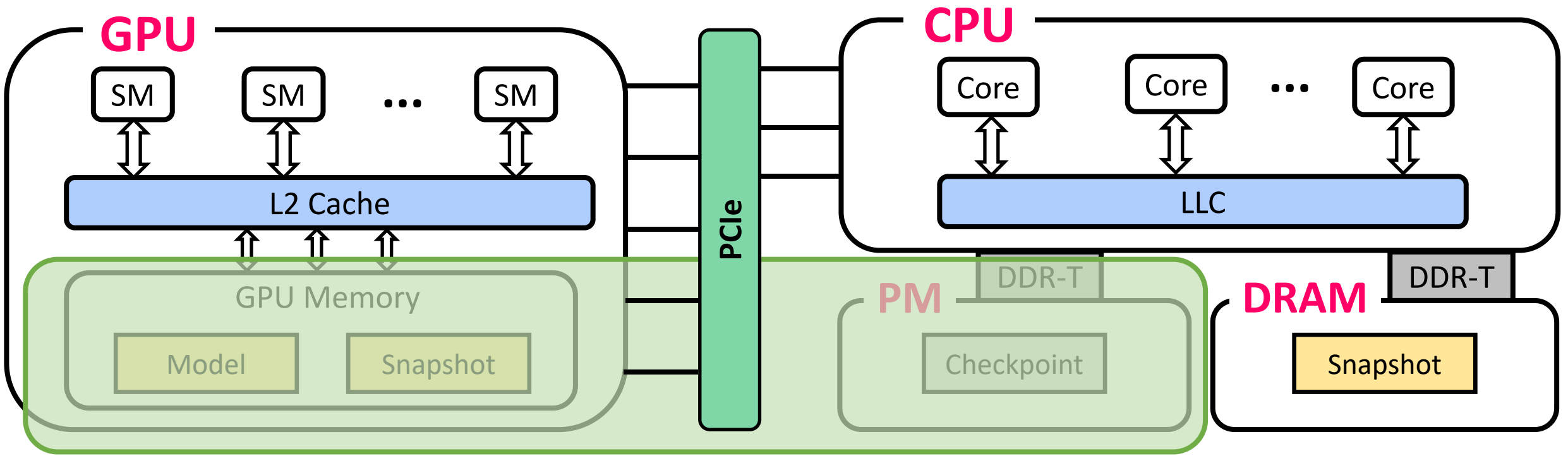




Efficient persistent memory management

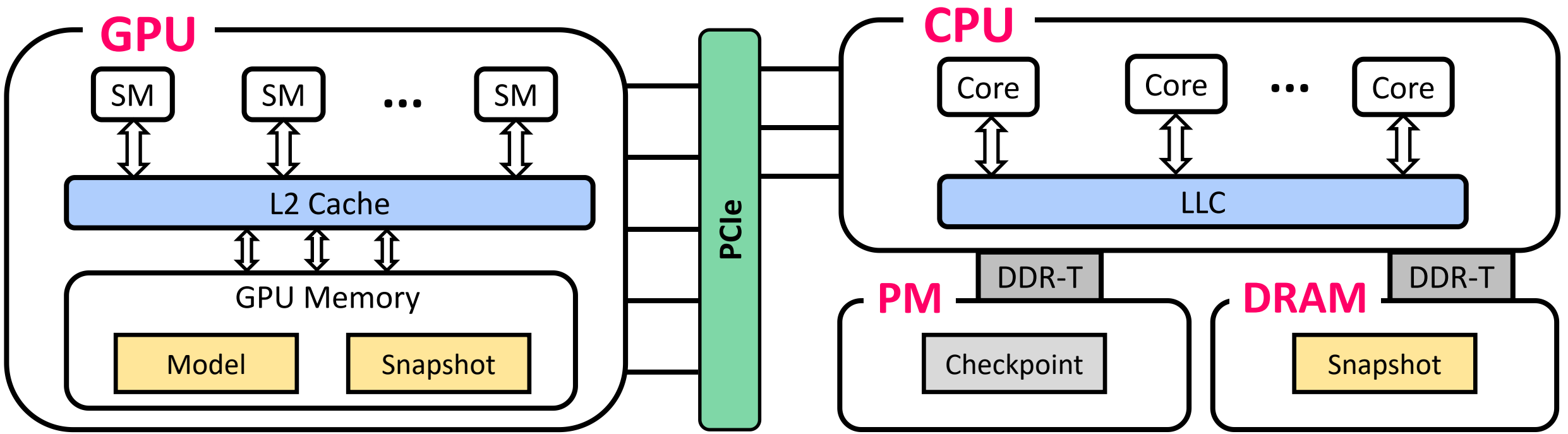


Efficient persistent memory management

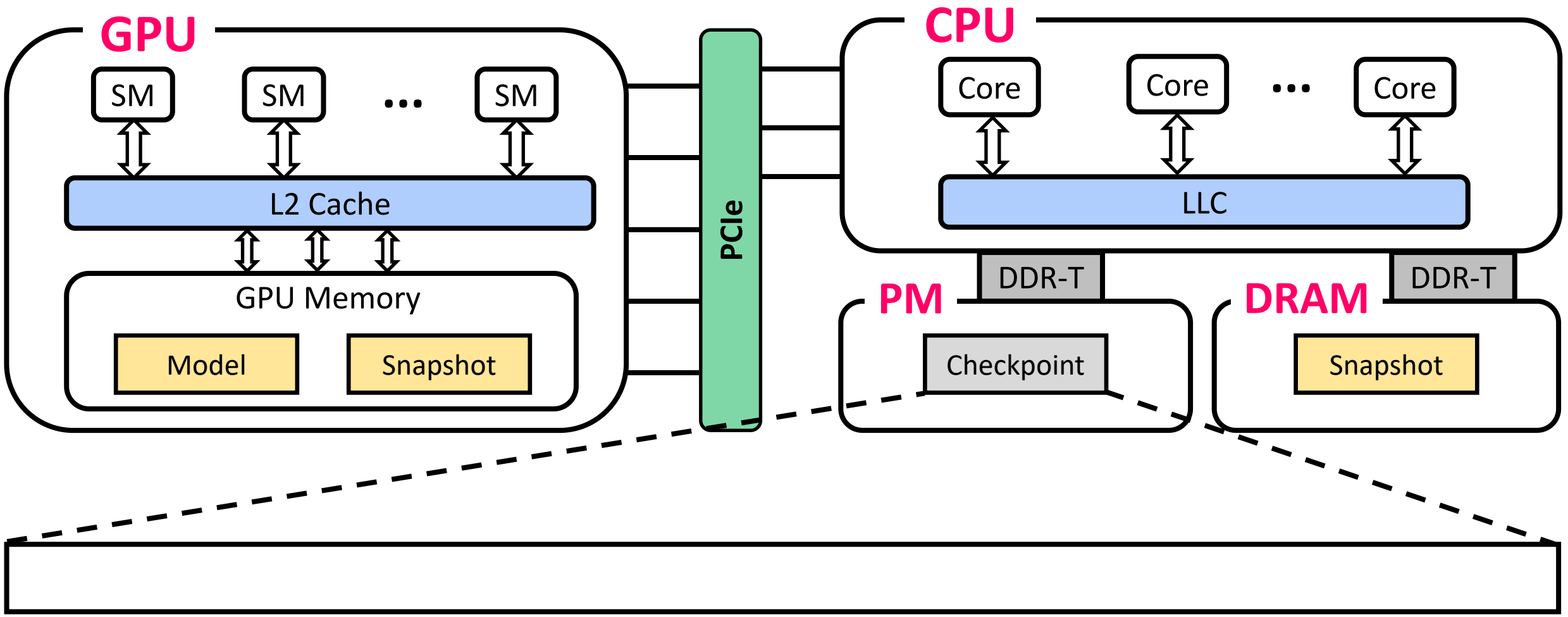


Unified virtual addressing (UVA)

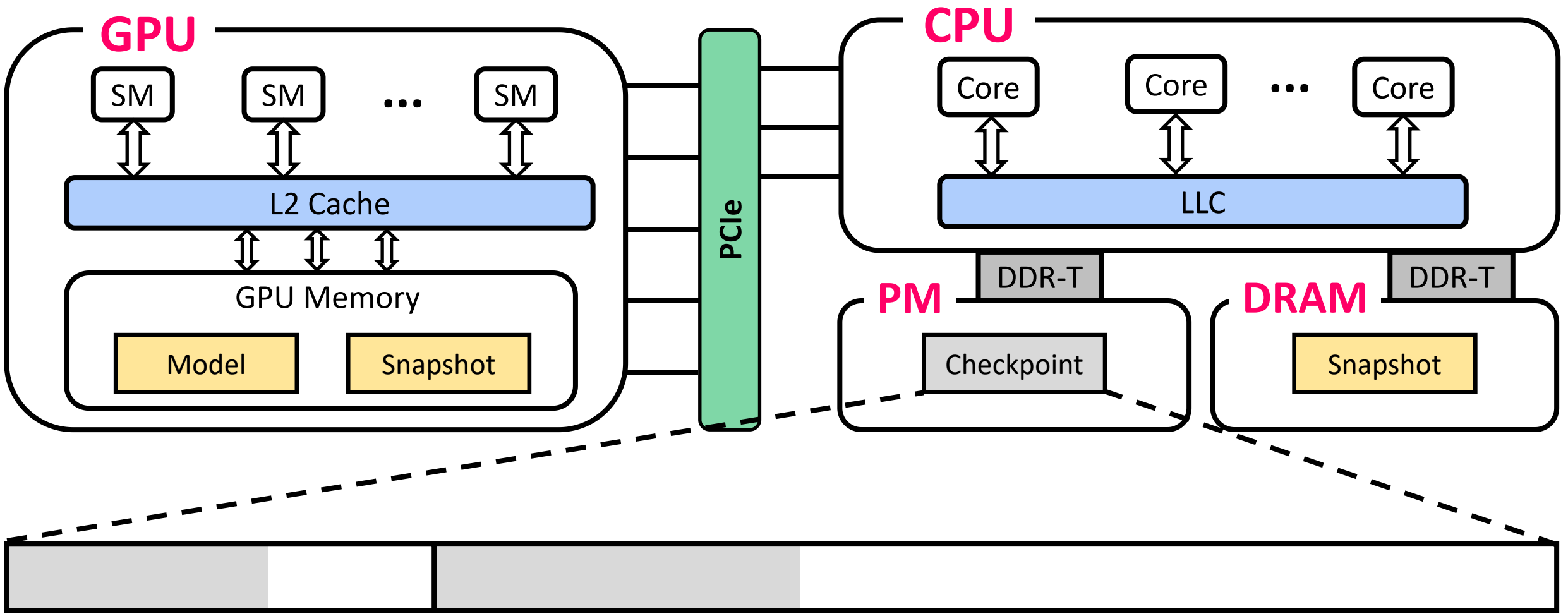
Efficient persistent memory management



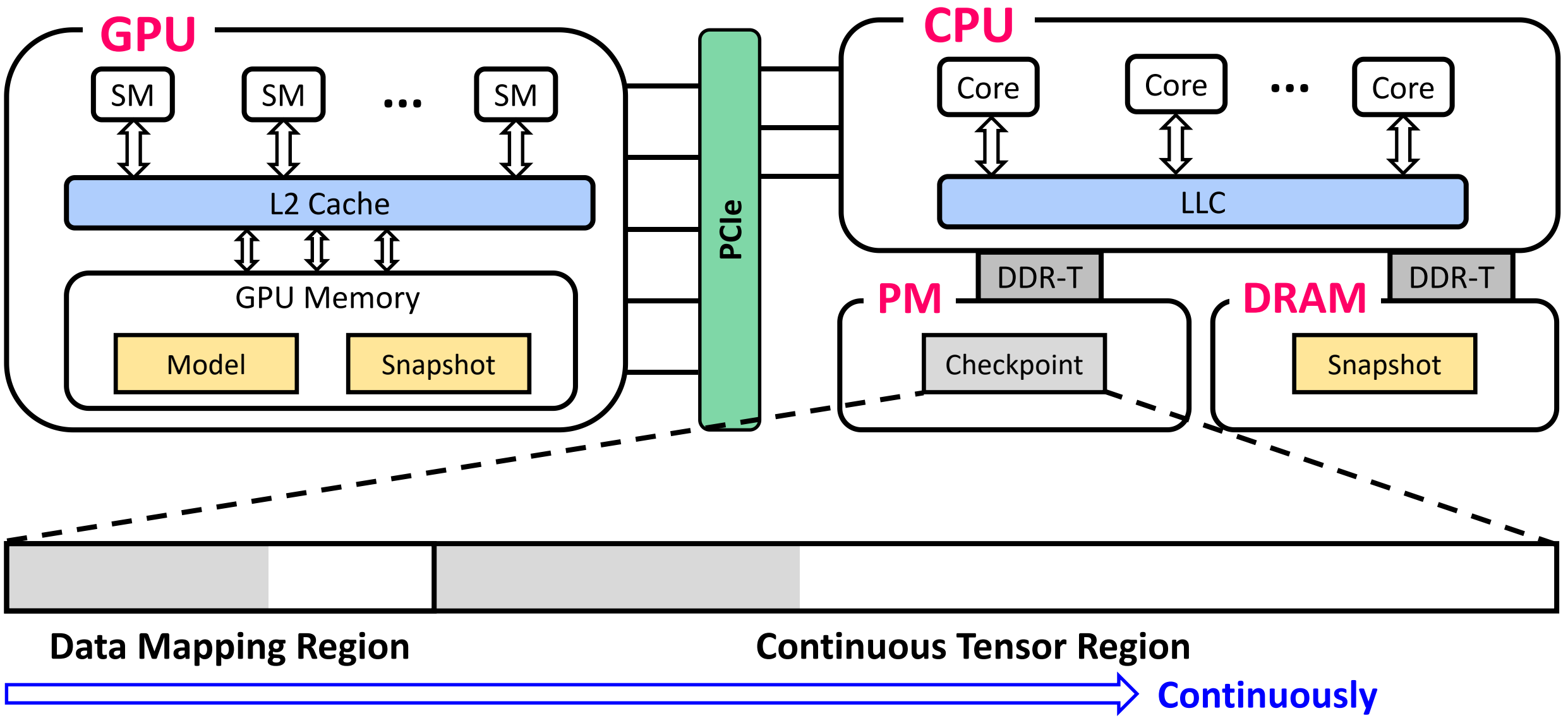
Efficient persistent memory management



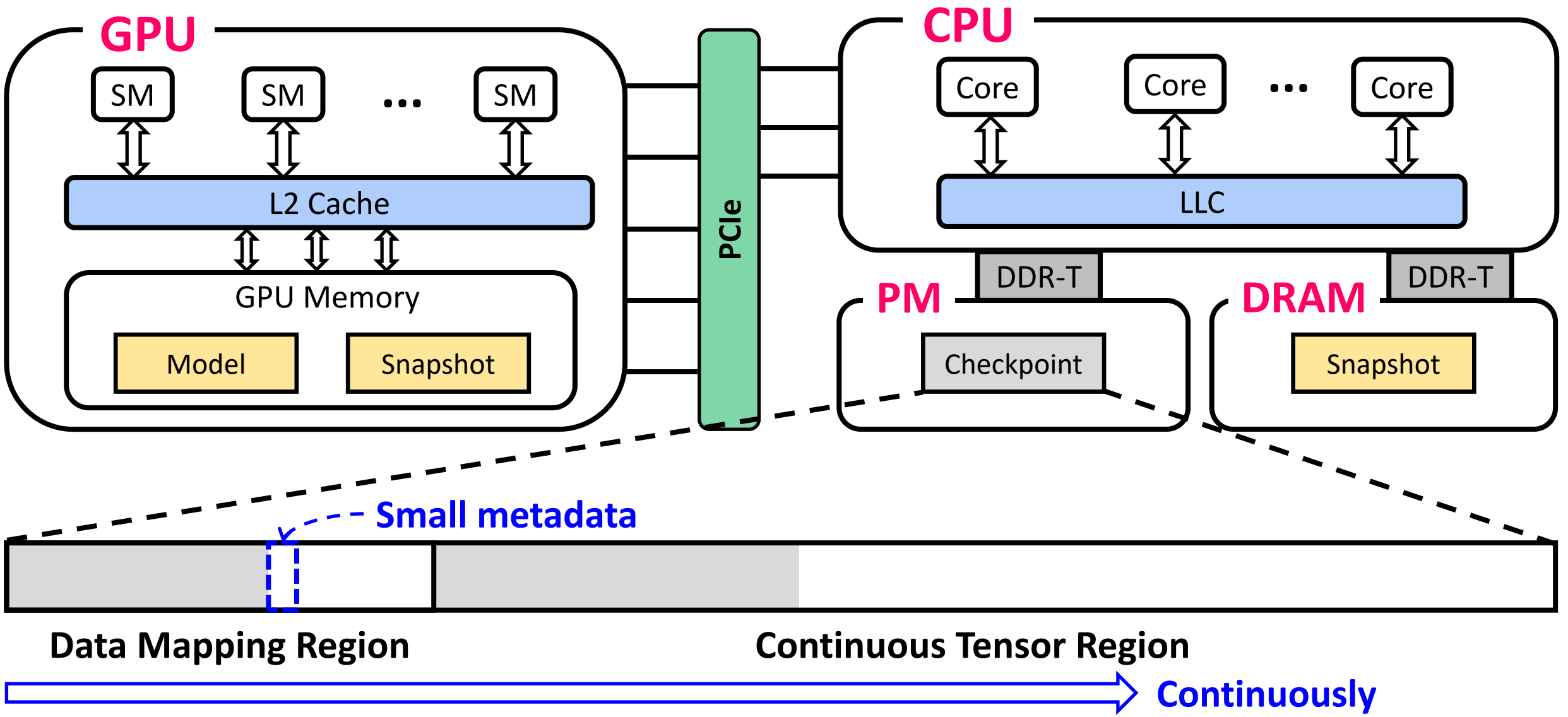
Efficient persistent memory management



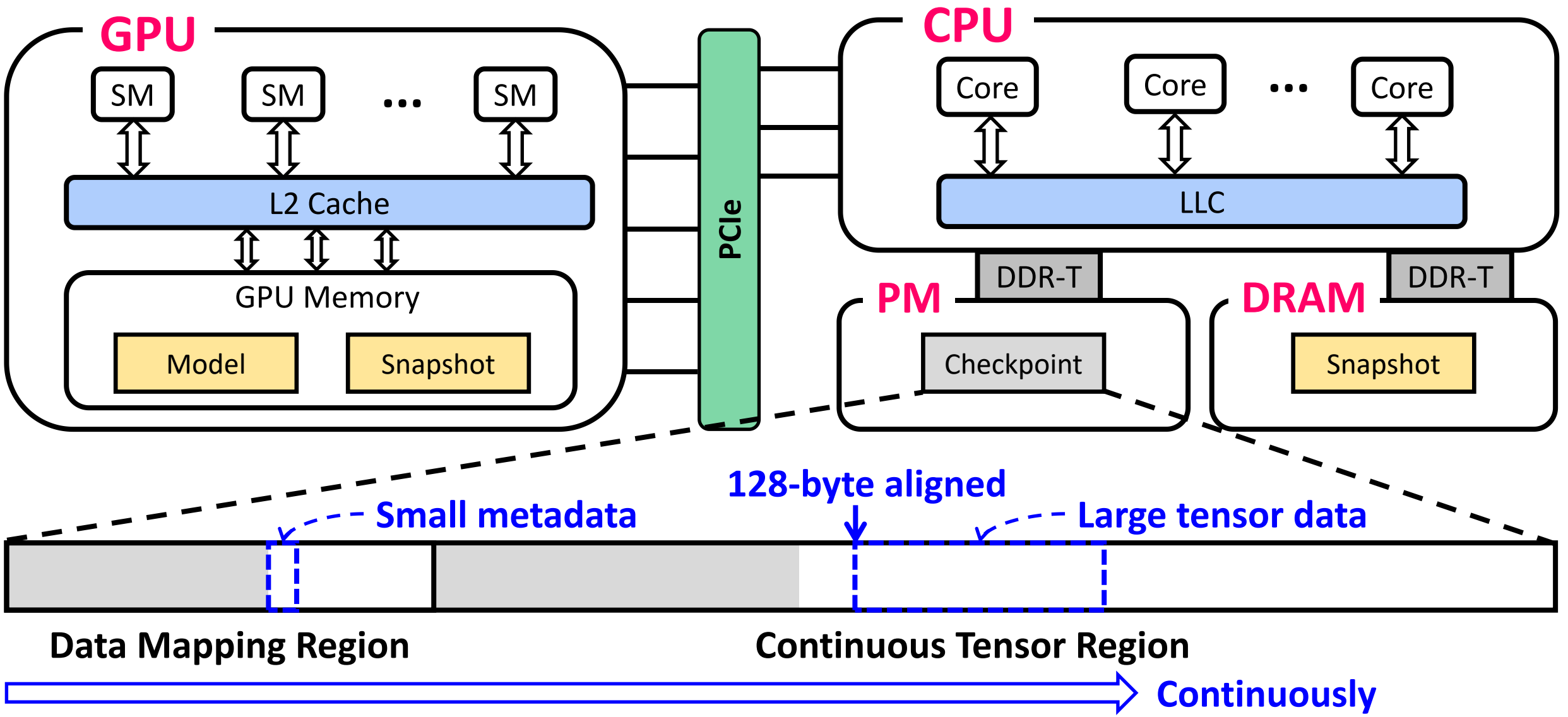
Efficient persistent memory management



Efficient persistent memory management



Efficient persistent memory management



Evaluation

Evaluation

- **Platform**
  - Three nodes connected via 100 Gbps Mellanox InfiniBand switch
- **DNN Models**
  - ResNet-18, VGG-16, Inception-V3, AlexNet, GPT-2, BERT
- **Comparisons**
  - CheckFreq, Pytorch

Evaluation

➢ Platform

- Three nodes connected via 100 Gbps Mellanox InfiniBand switch

➢ DNN Models

- ResNet-18, VGG-16, Inception-V3, AlexNet, GPT-2, BERT

➢ Comparisons

- CheckFreq, Pytorch

Sever Configuration

Machine	CPU	GPU	Memory	Storage	Network
3 nodes	Intel Xeon Gold 6230R, 26 cores	1 Tesla V100, 16GB	192GB DRAM, 6 X 128GB Intel Optane PM Modules	3.6TB HDD	100Gbps Mellanox InfiniBand Switch

Checkpointing Frequency

- Limit runtime overhead within 5%

Checkpointing Frequency

- Limit runtime overhead within 5%

Models	Checkpoint Size (MB)	Number of Iterations					
		LightCheck-G	LightCheck-C	LightCheck-D	LightCheck-disk	CheckFreq	torch.save
ResNet-18	90	1	1	1	7	20	102
VGG-16	1,056	6	6	6	64	146	904
Inception-V3	183	14	14	14	30	40	118
AlexNet	467	8	8	8	95	164	1,084
GPT-2	1,508	6	6	6	46	100	682
BERT	4,004	10	10	10	82	200	1,100

Checkpointing Frequency

- Limit runtime overhead within 5%

Models	Checkpoint Size (MB)	Number of Iterations					
		LightCheck-G	LightCheck-C	LightCheck-D	LightCheck-disk	CheckFreq	torch.save
ResNet-18	90	1	1	1	7	20	102
VGG-16	1,056	6	6	6	64	146	904
Inception-V3	183	14	14	14	30	40	118
AlexNet	467	8	8	8	95	164	1,084
GPT-2	1,508	6	6	6	46	100	682
BERT	4,004	10	10	10	82	200	1,100

LightCheck can achieve **frequent** checkpointing with **modest** runtime overhead

Up to 10X

Checkpointing Frequency

- Limit runtime overhead within 5%

Models	Checkpoint Size (MB)	Number of Iterations					
		LightCheck-G	LightCheck-C	LightCheck-D	LightCheck-disk	CheckFreq	torch.save
ResNet-18	90	1	1	1	7	20	102
VGG-16	1,056	6	6	6	64	146	904
Inception-V3	183	14	14	14	30	40	118
AlexNet	467	8	8	8	95	164	1,084
GPT-2	1,508	6	6	6	46	100	682
BERT	4,004	10	10	10	82	200	1,100

Checkpointing Frequency

- Limit runtime overhead within 5%

Models	Checkpoint Size (MB)	Number of Iterations					
		LightCheck-G	LightCheck-C	LightCheck-D	LightCheck-disk	CheckFreq	torch.save
ResNet-18	90	1	1	1	7	20	102
VGG-16	1,056	6	6	6	64	146	904
Inception-V3	183	14	14	14	30	40	118
AlexNet	467	8	8	8	95	164	1,084
GPT-2	1,508	6	6	6	46	100	682
BERT	4,004	10	10	10	82	200	1,100

Asynchronous layer-wise checkpointing **reduces** the runtime overhead

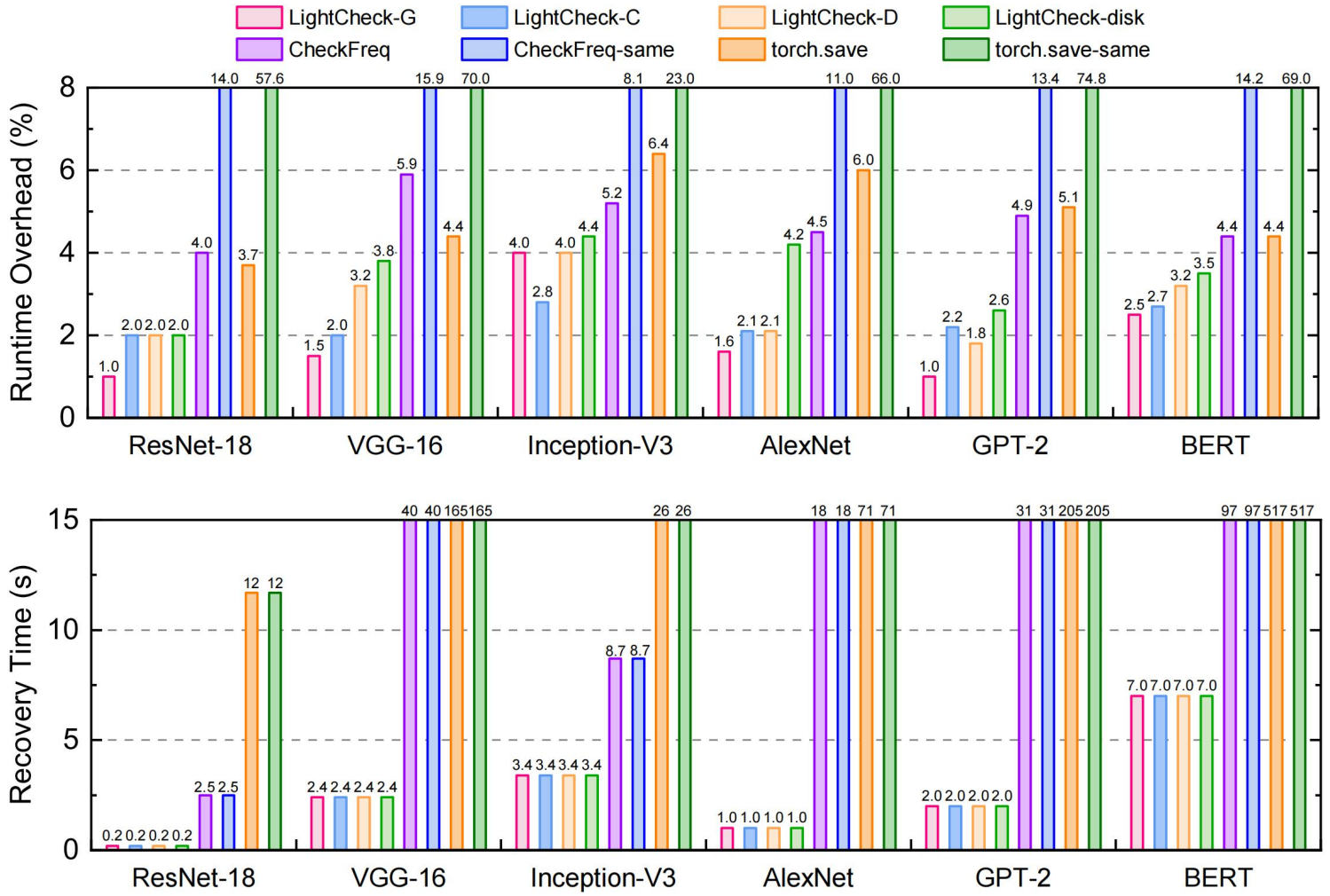
Up to 2X

Overall Performance

- With the aboved checkpointing frequency

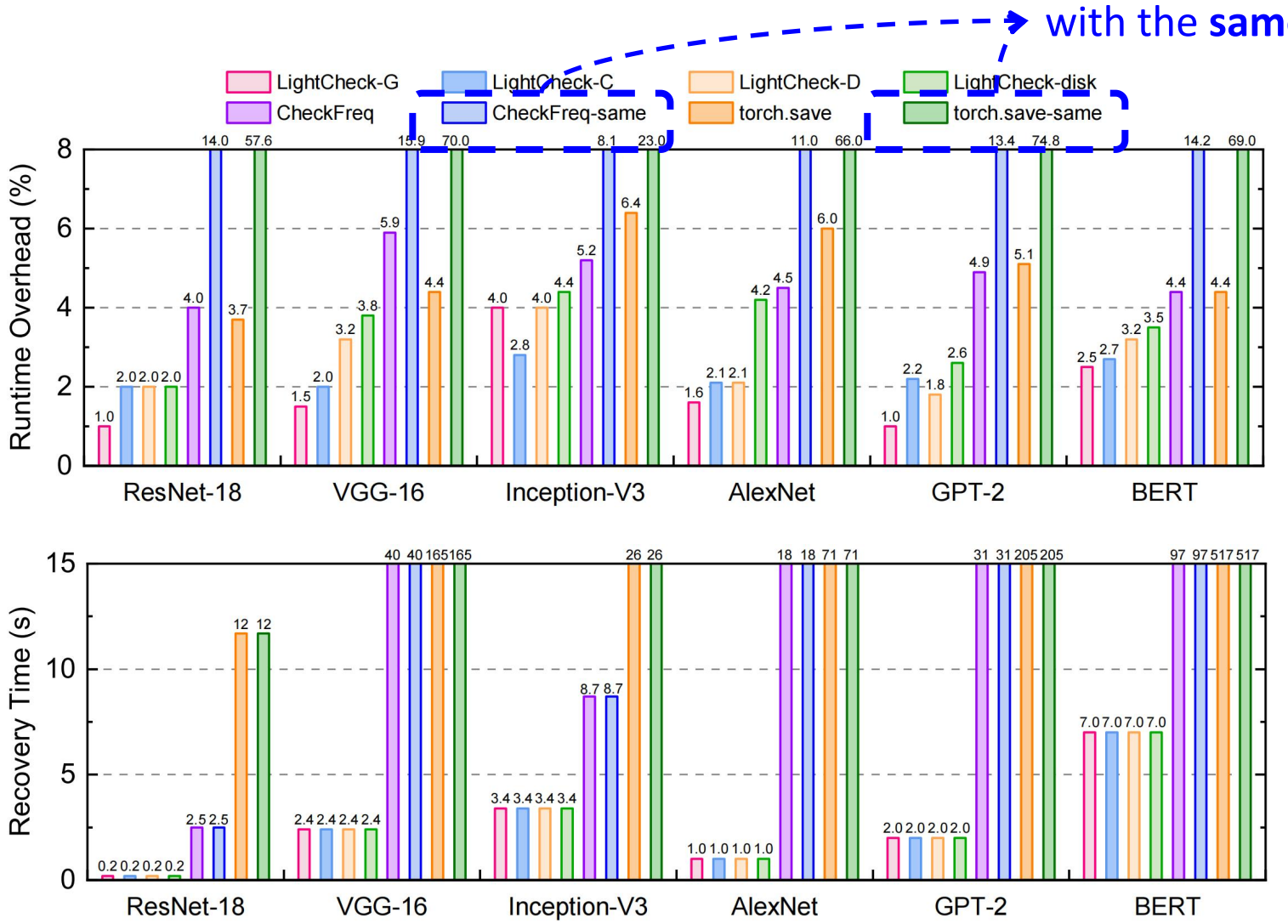
Overall Performance

- With the aboved checkpointing frequency



Overall Performance

- With the aboved checkpointing frequency

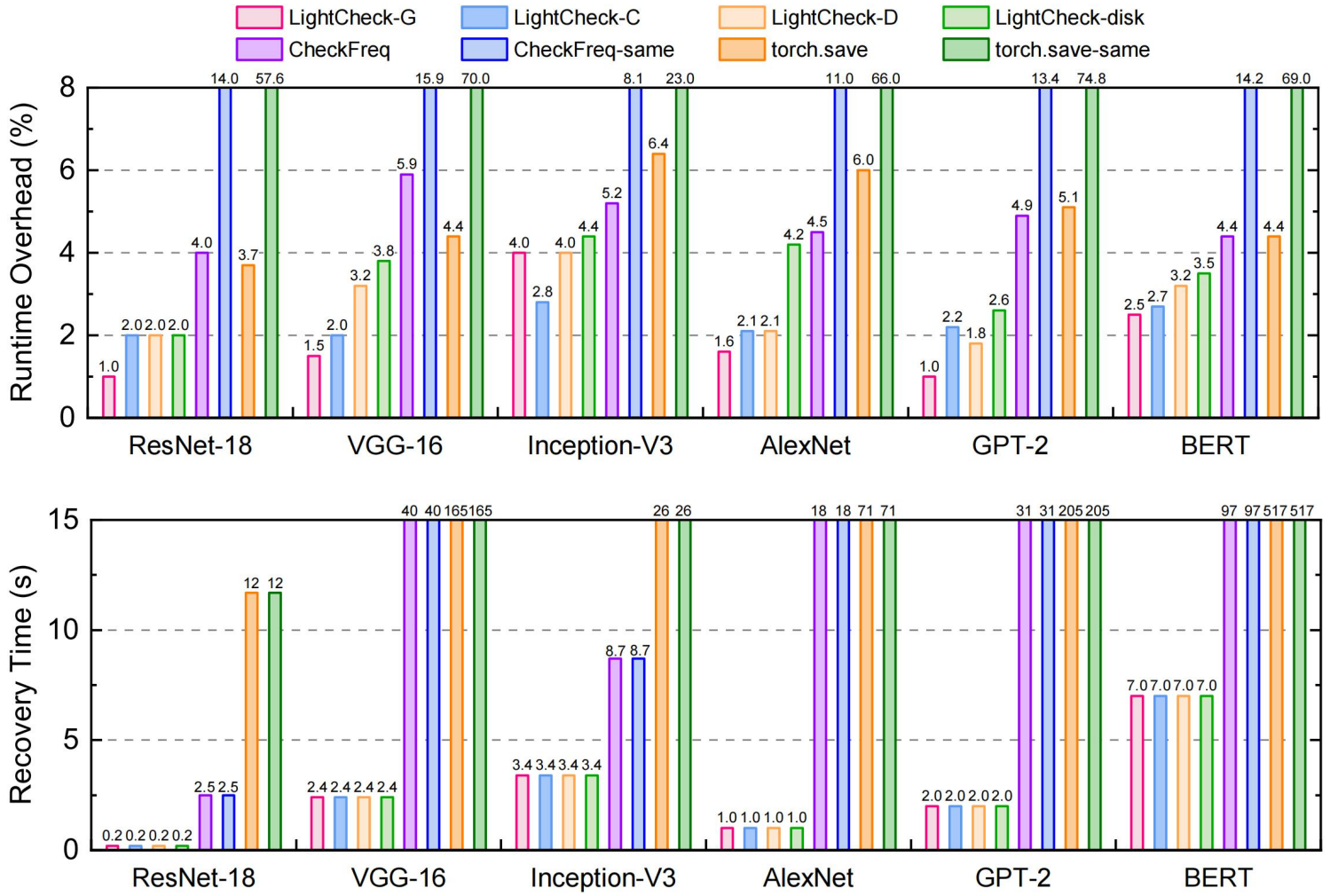


Overall Performance

- With the aboved checkpointing frequency

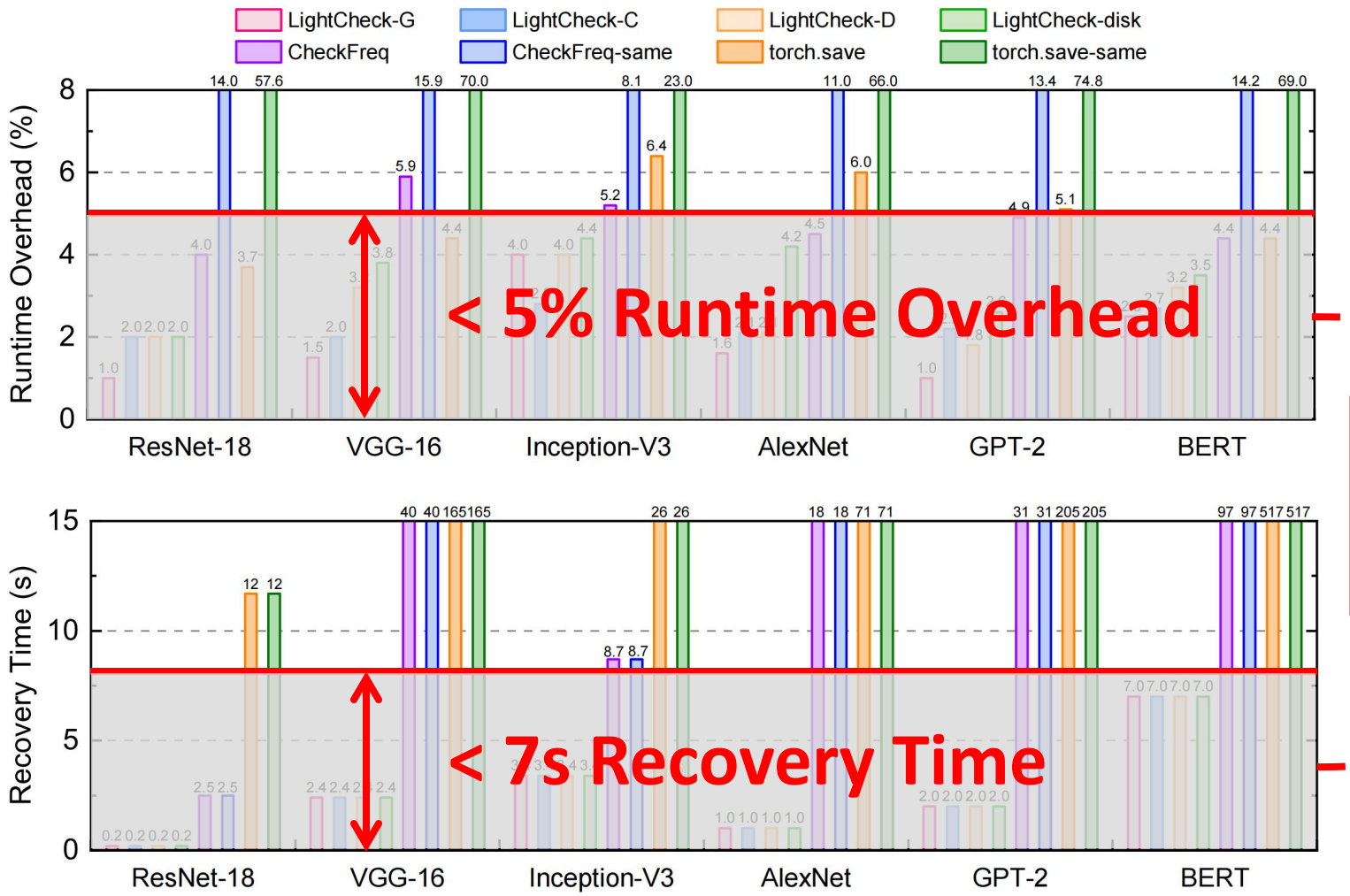
Overall Performance

- With the aboved checkpointing frequency



Overall Performance

- With the aboved checkpointing frequency



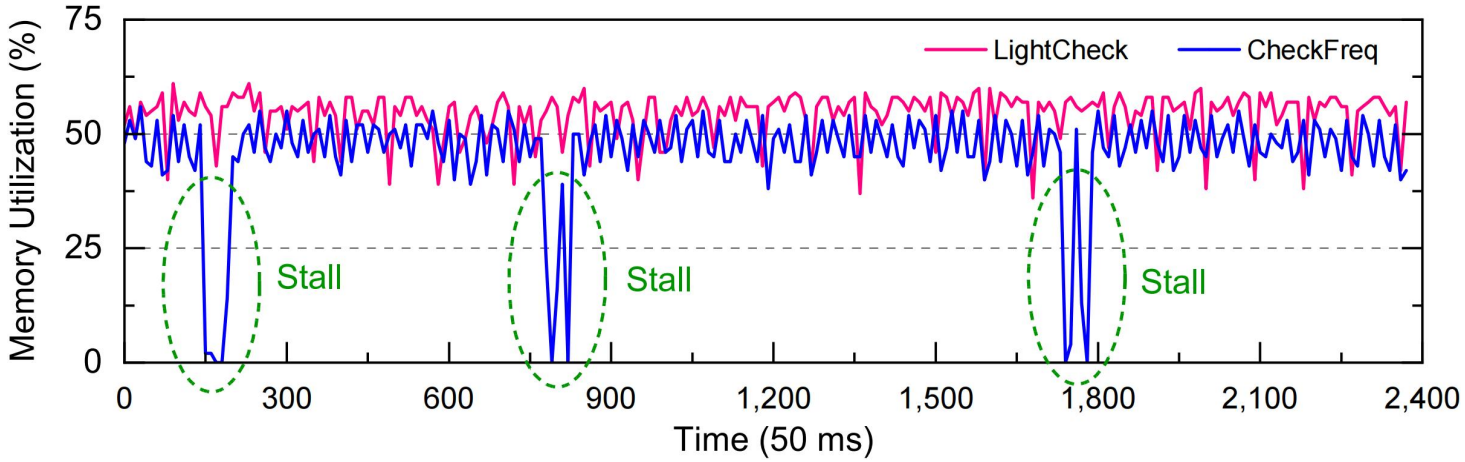
LightCheck provides **lower** recovery time and overhead than existing schemes

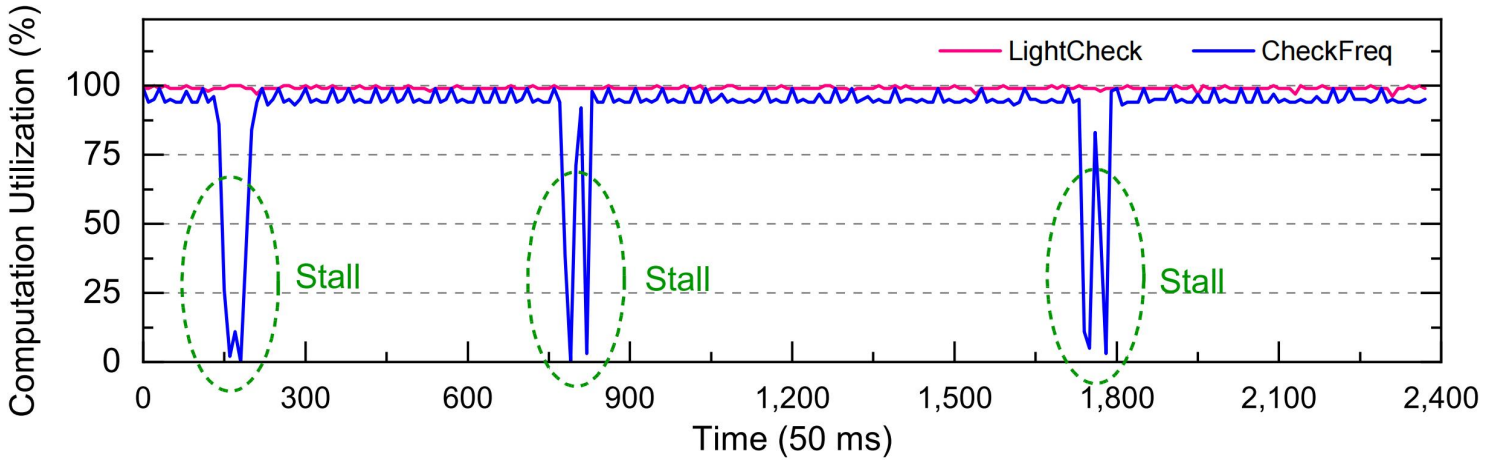
GPU Utilization

- Record the GPU utilization every 50 ms, VGG-16

GPU Utilization

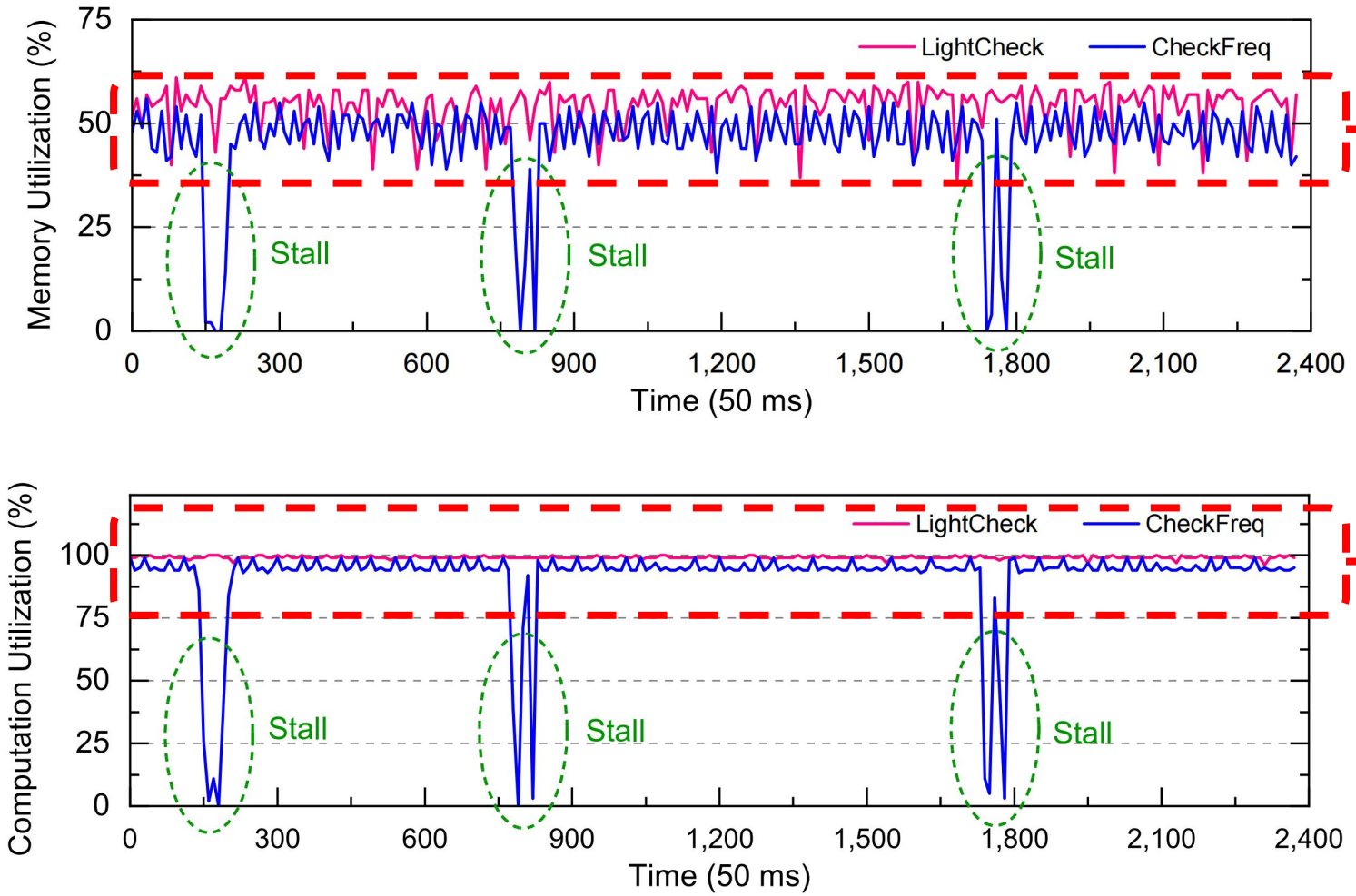
- Record the GPU utilization every 50 ms, VGG-16





GPU Utilization

- Record the GPU utilization every 50 ms, VGG-16



LightCheck **eliminates** training stall by leveraging find-grained pipelining

Conclusion

- **LightCheck: A cost-efficient checkpointing scheme for DNN training**
  - Asynchronous layer-wise checkpointing
  - Efficient persistent memory management
- More evaluation results and analysis are in the paper
- Available at: https://github.com/LighT-chenml/LightCheck.git

Conclusion

- **LightCheck: A cost-efficient checkpointing scheme for DNN training**
  - Asynchronous layer-wise checkpointing
  - Efficient persistent memory management
- More evaluation results and analysis are in the paper
- Available at: https://github.com/LighT-chenml/LightCheck.git

Thank you! Q&A