# NCHC Open Hackathon–Final Presentation

**Lab**: Nano-photonic and Micro-Optical System

**Graduate Institute**: Automation Technology

---

NTUT Birdsong

# NTUT BirdSong

## Member

## Mentor

NTUT
Jin-Jia Hu

NTUT
Yang-Yu Ou

NTUT
Xin-He Chen

NVDIA
Virginia Chen

NTUT
Yi-Yang Syu

NTUT
Yung-Yu Chen

NTUT
Shih-Cheng Ma

NVDIA
Iven Fu

# Progress and Goals

- **Progress**
  - Rewrote the code with Transformer Engine.
  - Achieved a 21% reduction in computing time.
- **Goals**
  - Accomplished an acceleration result.

# Application Name

- **Problem the team is trying to solve.**

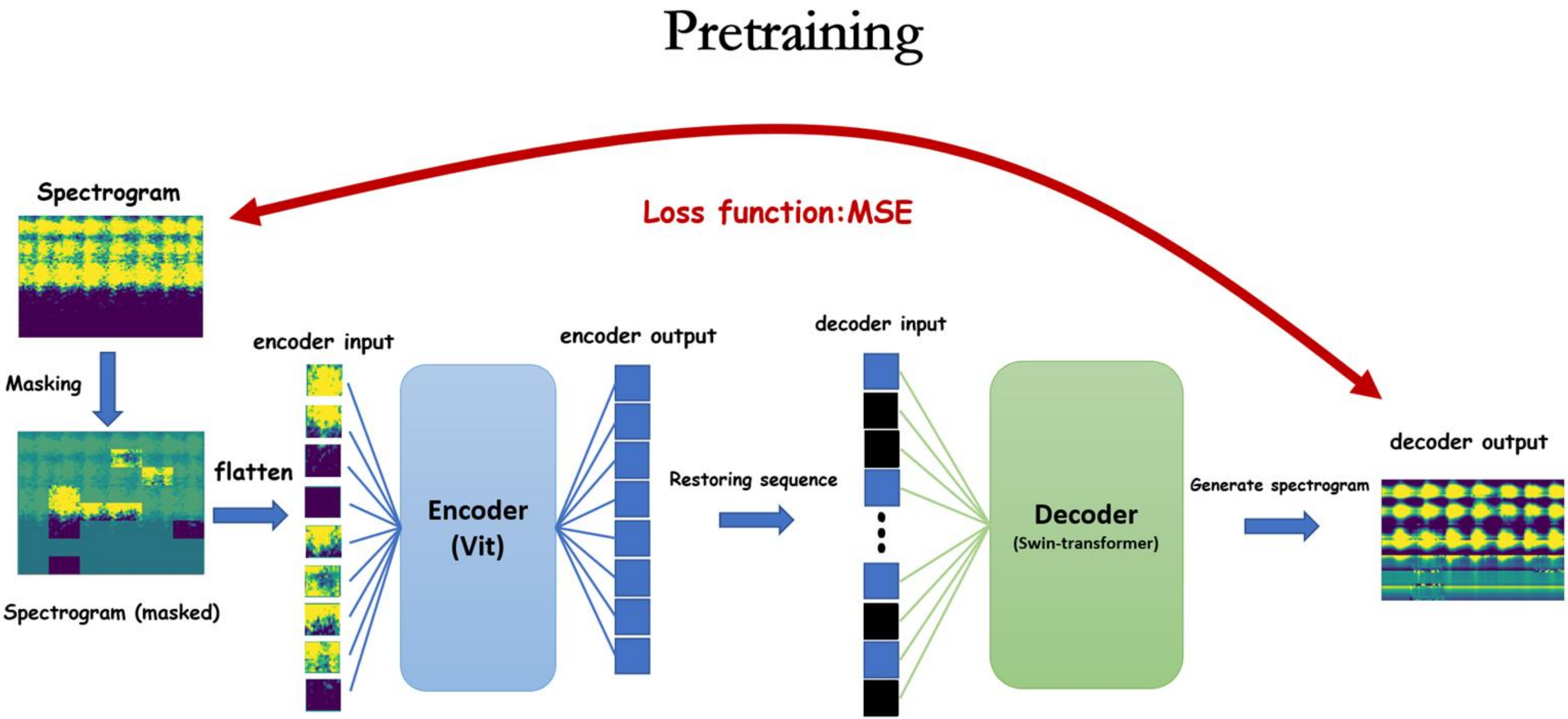
  Reduce pretraining time
- **Scientific driver for the chosen algorithm.**

  Audio-MAE
- **What’s the algorithmic motif?**

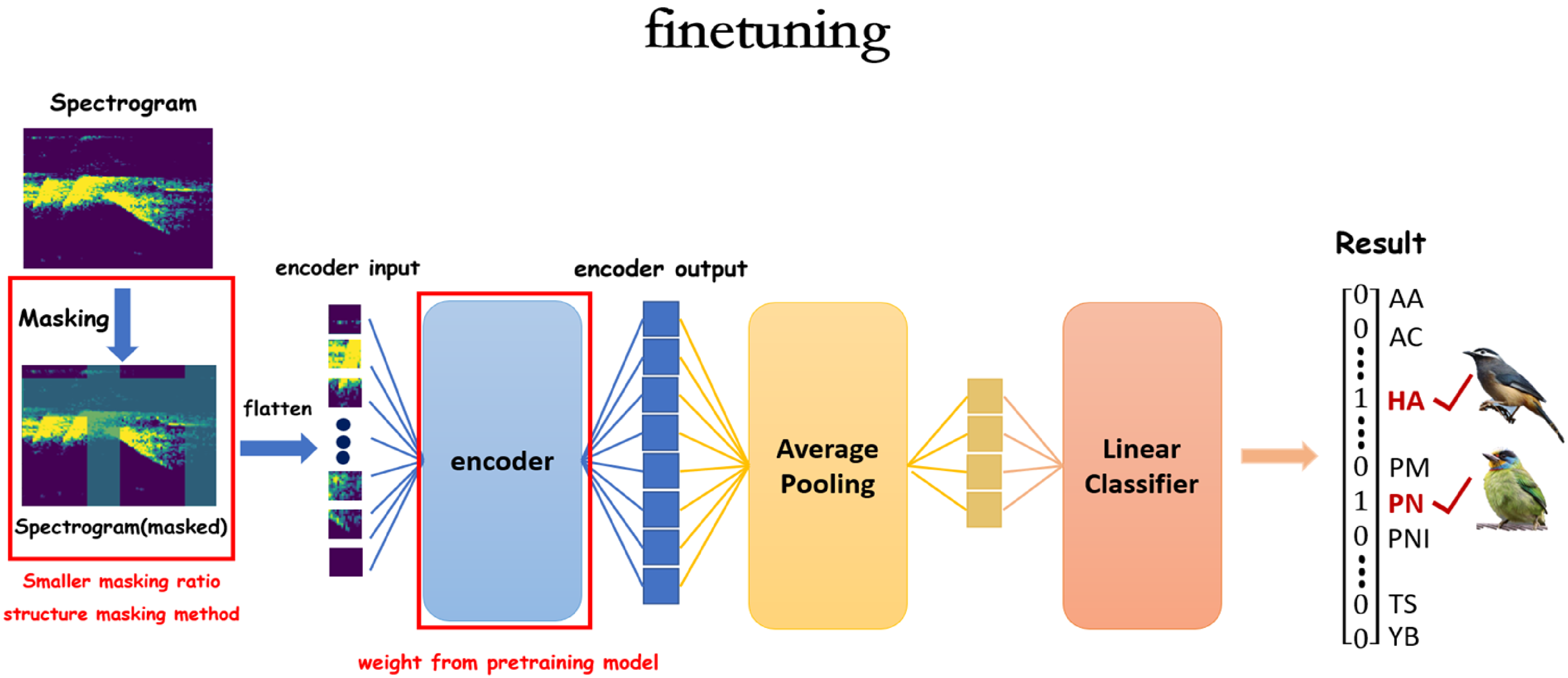
  swin-transformer,VIT
- **What parts are you focused on?**

  pre-train

# Model Architecture[1]



# Model Architecture[1]



# Evolution and Strategy

- **What was your goal for coming here?**

  Expanding knowledge,Academic exchange,Accelerating computation

- **What was your initial strategy?**

  Use NEMO framework

- **How did this strategy change?**

  NEMO does not include the Swin-Transformer architecture, so we handle it by segmenting the process with Transformer Engine

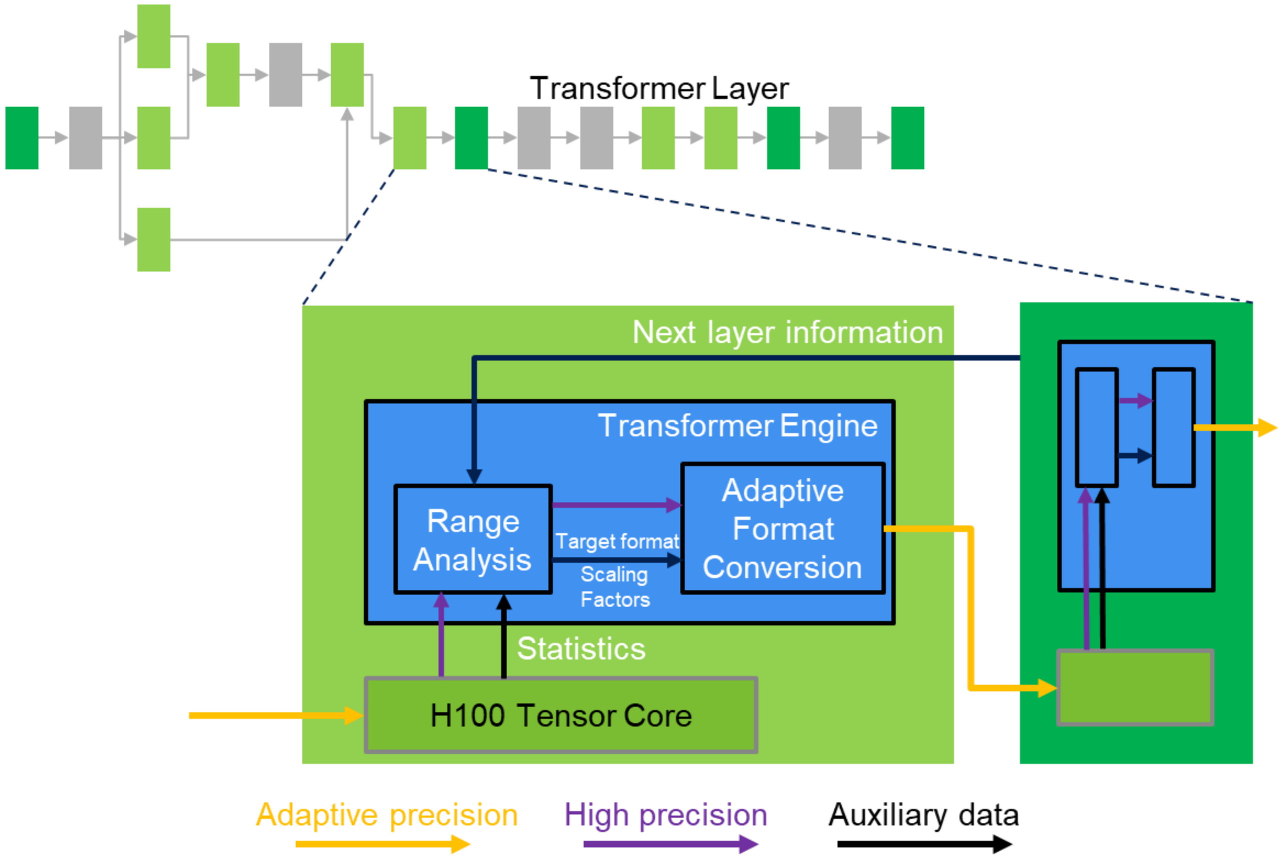
# Acceleration Technology[5]

**Data Type Optimization**: Transformer Engine uses mixed-precision computation (e.g., using FP8 or FP16 instead of traditional FP32), which significantly boosts computational efficiency while reducing GPU memory requirements.

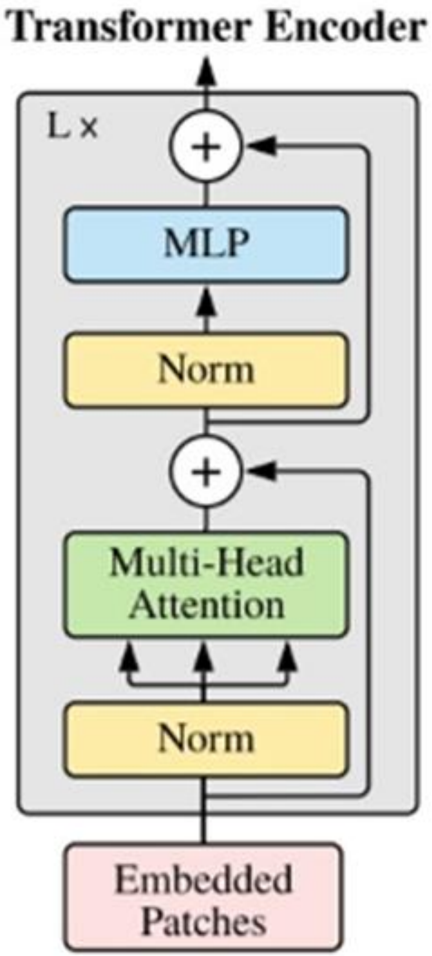
**Computation Graph Optimization**: Transformer Engine optimizes the model's computation graph to ensure the best allocation of computational resources, avoiding bottlenecks during the process.

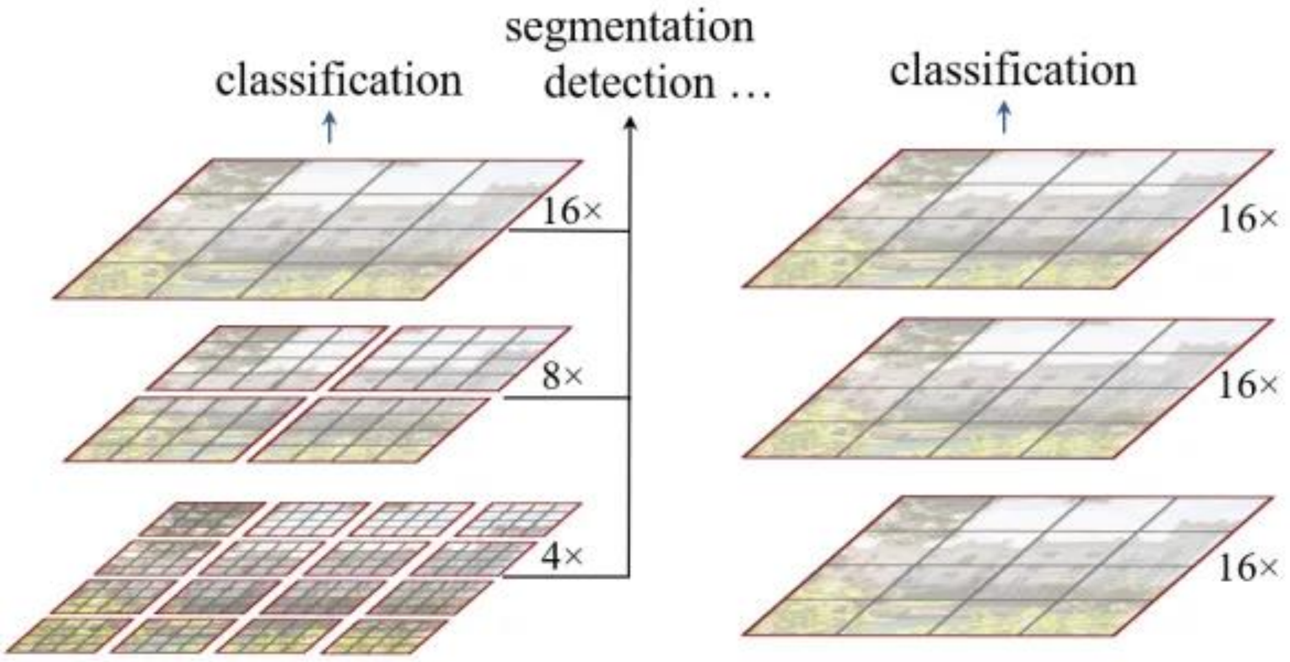
**Dedicated Acceleration Hardware**: It leverages the Tensor Cores of NVIDIA A100 and H100 GPUs, which are specialized for accelerating matrix operations, further enhancing both inference and training speed of deep learning models.

**Memory Management**: Transformer Engine efficiently manages GPU memory, particularly in large-scale model training, avoiding memory bottlenecks and reducing data transfer delays.



# Model Architecture[2,3]

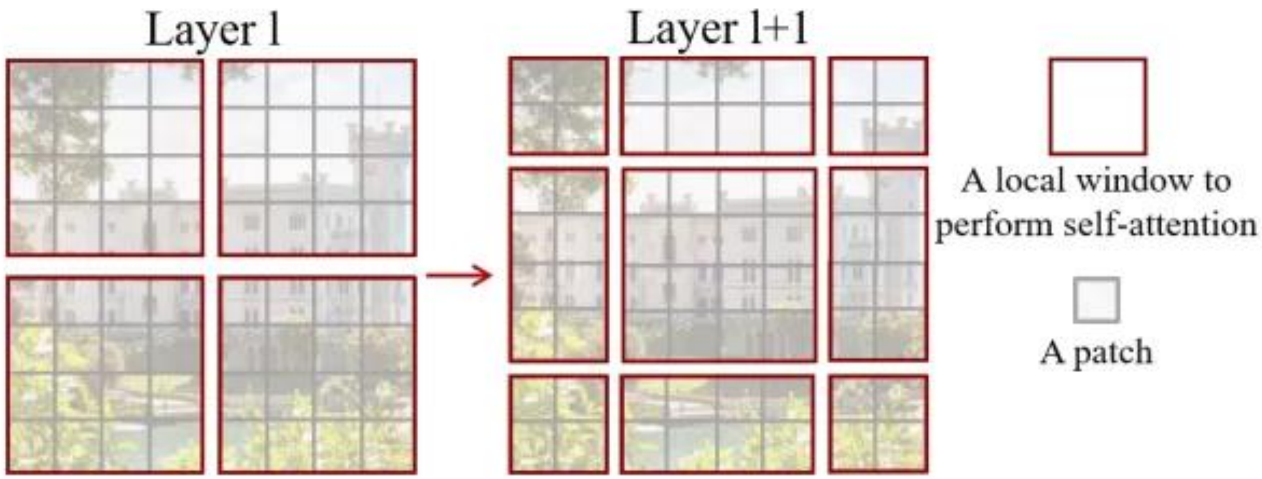


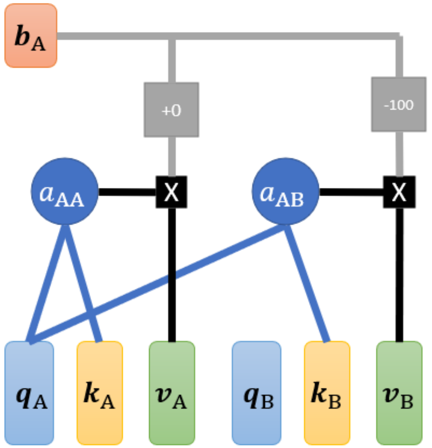



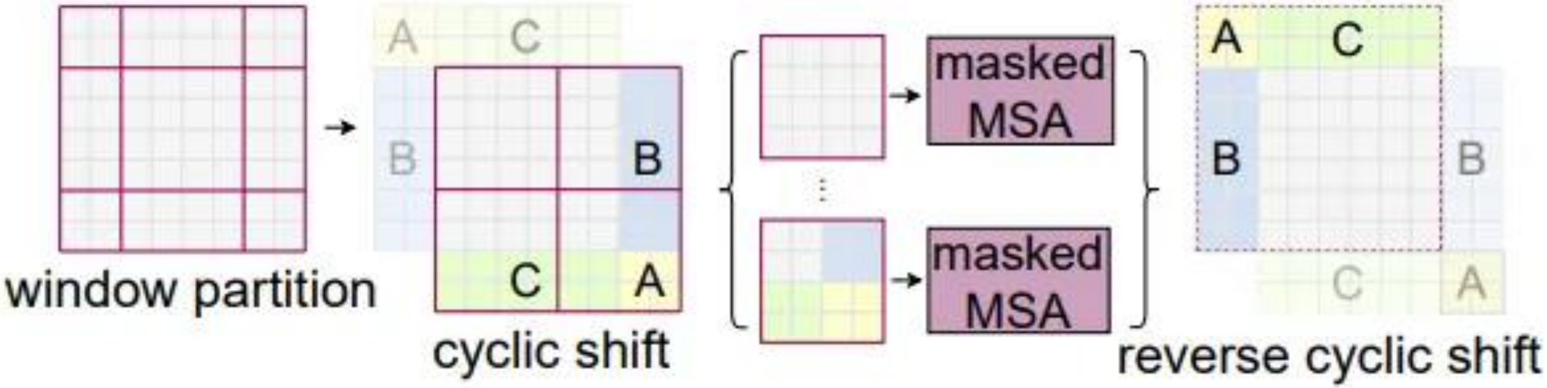

(a) Swin Transformer (ours)

(b) ViT

# Model Architecture[3]



# Results and Final Profile

- **What were you able to accomplish?**

  The software optimization achieved a 1.3x acceleration, resulting in a total speed-up of 3.6x

- **Show multi-core vs. GPU numbers**

  Use two A100 GPUs

- **What did you learn?**

  Use Transformer Engine architecture to accelerate model

  Use Nsight system to analyze GPU utilization

  Understand the limitation of different GPUs

# Accelerate Result

## 【Acceleration performance of different GPUs】

Dataset: **6M**, model: **base**, epoch:1

| GPU | Batch size | TE | Spend time | time(s) / Iter | Iters | Accelerate rate |
|---|---|---|---|---|---|---|
| 4090 | 64 | False | 2:04:11 | 0.0811 | 91881 | 105 % |
| | | True | 1:57:43 | 0.0769 | 91881 | |
| A100 | 512 * 2 | False | 43:10 | 0.4512 | 5742 | 127 % |
| | | True | 33:56 | 0.3546 | 5742 | |

# Accelerate Result

## 【Final acceleration results】

| GPU | batch size | TE | Spend time | Accelerate rate |
|---|---|---|---|---|
| A100 | 512 * 2 | True | 33:56 | 364 % |
| 4090 | 64 | False | 2:04:11 | |

# Final Thoughts

- **Was this Open Hackathon worth it?**

  This event was very beneficial for our team, helping us quickly learn new technologies, solve problems, and improve collaboration.

- **Will you continue development?**

  Yes, we will continue to dive deeper into the research and further optimize the speed, for example, using NEMO.

- **Next steps, future plans.**

  We will not only optimize our model's speed using TE but also leverage NEMO to further enhance its performance.

- **What sustained resources or support will be critical for your work after the event?**

  We will also need access to GPUs similar to those used in this event to ensure our model runs faster

# Acknowledgement

- Mentors
- NCHC Open Hackathon

# Reference


**[1]** Huang, P.Y., Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., Feichtenhofer, C., 2022. Masked autoencoders that listen. arXiv. https://doi.org/10.48550/ arXiv.2207.06405, 2207.06405v3.

**[2]** A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS’17. USA: Curran Associates Inc., 2017, pp. 6000–6010.

**[3]** Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. IEEE, 2021, pp. 9992–10 002.

**[4]** A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020.

**[5]** https://github.com/NVIDIA/TransformerEngine

# Thank you for your attention.