Understanding Swin Transformer and Comparison with ViT

Swin Transformer Architecture

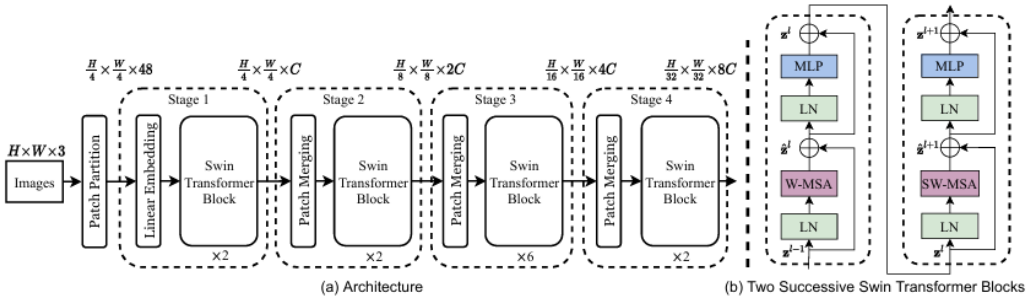
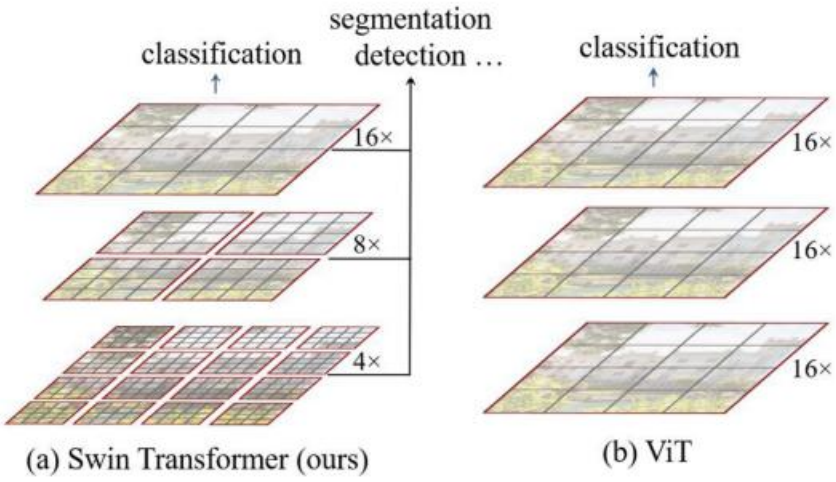


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

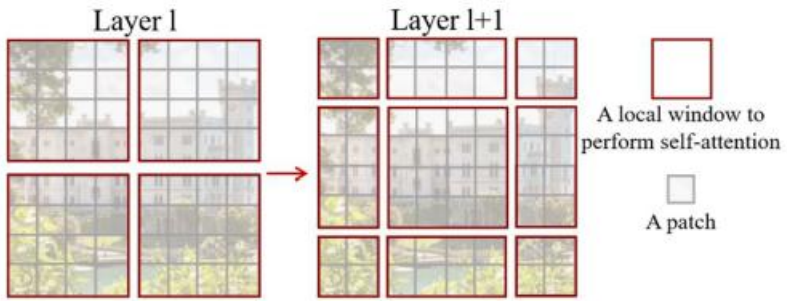
The Swin Transformer introduces a hierarchical design that processes input images in multiple stages. This structure enables multi-scale feature representation, which is essential for visual tasks such as detection and segmentation.

The input image is first divided into non-overlapping 4×4 patches. Each patch is linearly projected into a fixed-length embedding. These embeddings are passed through a series of Swin Transformer blocks organized into four sequential stages. At each stage, the spatial resolution of the feature map is reduced, while the number of channels is increased, forming a hierarchical feature pyramid.



Each Swin block contains window-based multi-head self-attention (W-MSA) layers. Instead of computing attention globally across the entire image (as in ViT), W-MSA operates locally within non-overlapping windows, significantly reducing computational cost from $O(N^2)$ to $O(N)$, where N is the number of patches.

To facilitate information exchange between neighboring windows, the architecture introduces shifted windows in alternating transformer layers. This technique, known as SW-MSA (Shifted Window MSA), enables cross-window communication without increasing the complexity of the attention mechanism.



Rather than using absolute positional encodings, Swin introduces a learnable relative position bias within each window. This design enhances translation invariance and is more naturally aligned with visual data.

Comparison: Swin Transformer vs ViT

Feature	ViT	Swin Transformer
Structure	Flat, uniform resolution	Hierarchical, multi-stage
Attention	Global self-attention	Local window attention with shifted windows
Computational Complexity	$O(N^2)$	$O(N)$
Patch Embedding	Fixed-size patches	Initial 4×4, then merged
Positional Encoding	Absolute	Relative positional bias
Feature Scaling	No multiscale support	Multiscale via patch merging
Image Size Flexibility	Limited	High — scalable to high resolutions
Object Detection	Needs FPN or added heads	Native support via hierarchical outputs
Semantic Segmentation	Requires additional modules	Integrated output compatible with decoders
Performance on ImageNet (Top-1)	~77.9% (ViT-B/16)	86.4% (Swin-B)
COCO Detection (Box AP)	~45–50 AP	58.7 AP
ADE20K Segmentation (mIoU)	~47 mIoU	53.5 mIoU

How Swin Transformer is Better than ViT

The Swin Transformer improves on the Vision Transformer (ViT) in several important ways, particularly for high-resolution and dense prediction tasks. One of the major advantages is computational efficiency. While ViT computes global self-attention, which has quadratic complexity with respect to the number of patches, Swin restricts attention to fixed-size windows, reducing the complexity to linear. This makes Swin significantly more scalable when processing large images. Another key improvement is Swin's hierarchical structure, which progressively merges patches to create multi-scale feature maps. This enables Swin to naturally extract low- to high-level features across its four stages, much like CNNs, whereas ViT operates at a single scale and lacks this capability. Swin is also better suited for tasks like object detection and semantic segmentation because its architecture produces multi-resolution outputs directly compatible with modern vision heads. In terms of performance, Swin consistently outperforms ViT across standard benchmarks. For example, Swin-B achieves 86.4% top-1 accuracy on ImageNet-1K, 58.7 box AP on COCO, and 53.5 mIoU on ADE20K, surpassing the original ViT models in all cases. Moreover, Swin avoids the need for absolute positional encodings by using learnable relative position bias within each attention window, which is simpler and better aligned with image data. Finally, the shifted window mechanism introduced in Swin allows for cross-window information flow without requiring full global attention, providing an elegant balance between local efficiency and global context. These design choices make Swin Transformer a more versatile and powerful backbone for a wide range of computer vision tasks compared to ViT.

Benefits of Swin Transformer for Vision Tasks

The hierarchical structure allows the Swin Transformer to generate feature maps at multiple resolutions. This makes it more suitable for object detection and segmentation than ViT, which operates on a flat sequence of patches. The window-based attention ensures linear computational complexity, while shifted windows ensure coverage across the image space.

The architecture works well as a drop-in replacement for CNNs in tasks that require dense predictions, such as segmentation. Swin achieves state-of-the-art results on benchmarks like ImageNet (classification), COCO (detection), and ADE20K (segmentation).

Result-Based Comparison: Swin Transformer vs ViT on CIFAR-10

Swin Transformer Results (from your log):

- **Final Test Accuracy**: **90.46%** (at epoch 200)
- **Training Accuracy**: **67.89%**
- **Final Test Loss**: **0.3056**

Vision Transformer (ViT) Results (from your earlier log):

- **Final Test Accuracy**: **91.65%** (at epoch 200)
- **Training Accuracy**: **70.35%**
- **Final Test Loss**: **0.2875**

Direct Performance Comparison

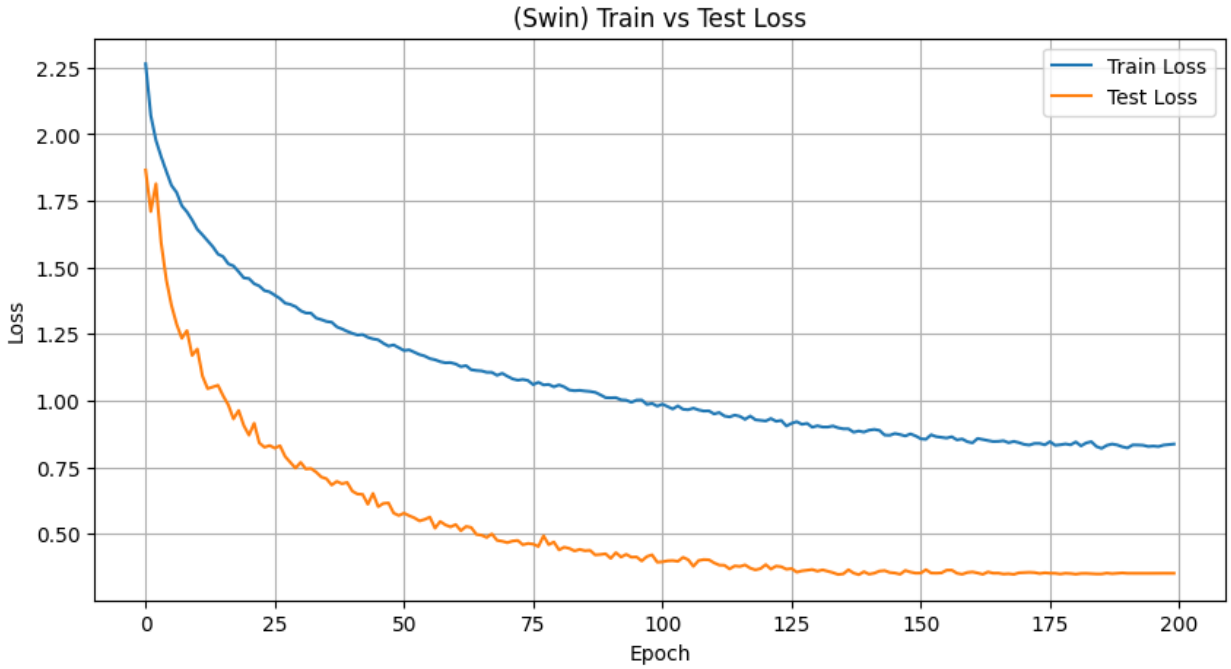
Metric	Swin Transformer	Vision Transformer
Test Accuracy	**90.46%**	**91.65%**
Training Accuracy	67.89%	70.35%
Test Loss	0.3056	**0.2875**

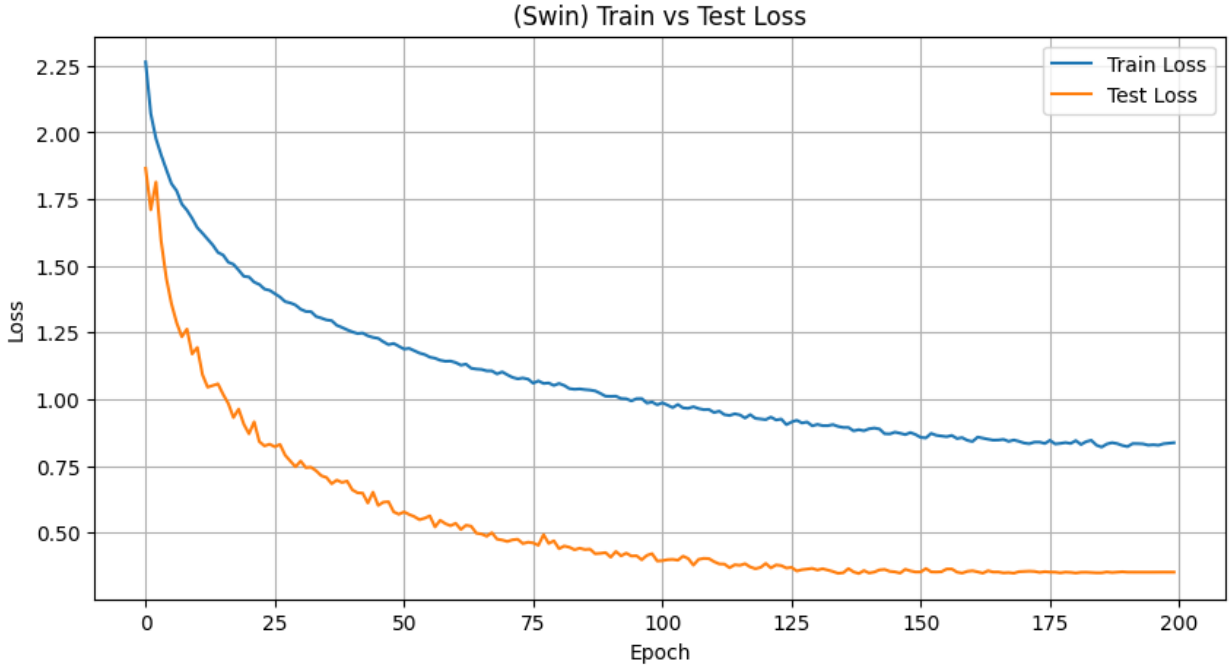
Analysis

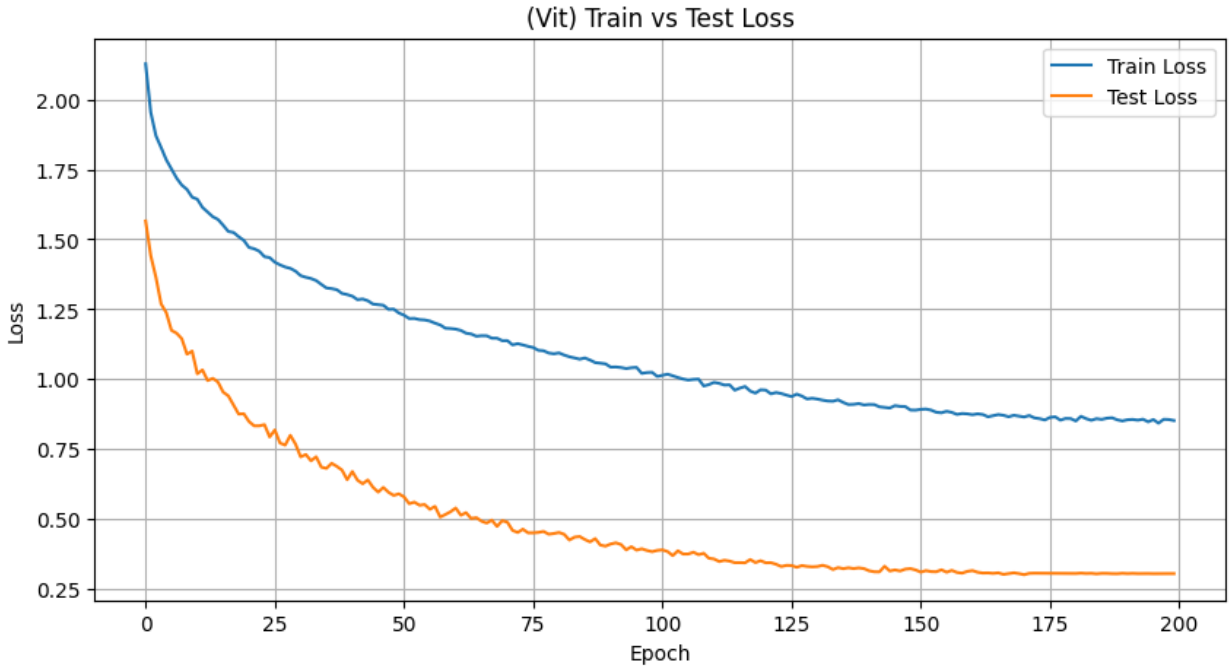
- **ViT performs better** than Swin Transformer on all key metrics in your training logs.
- **Swin** shows more gradual improvement, indicating stable but slower convergence.
- **ViT** reaches higher accuracy and lower loss within the same number of epochs (200), making it more effective **in your specific training setup**.

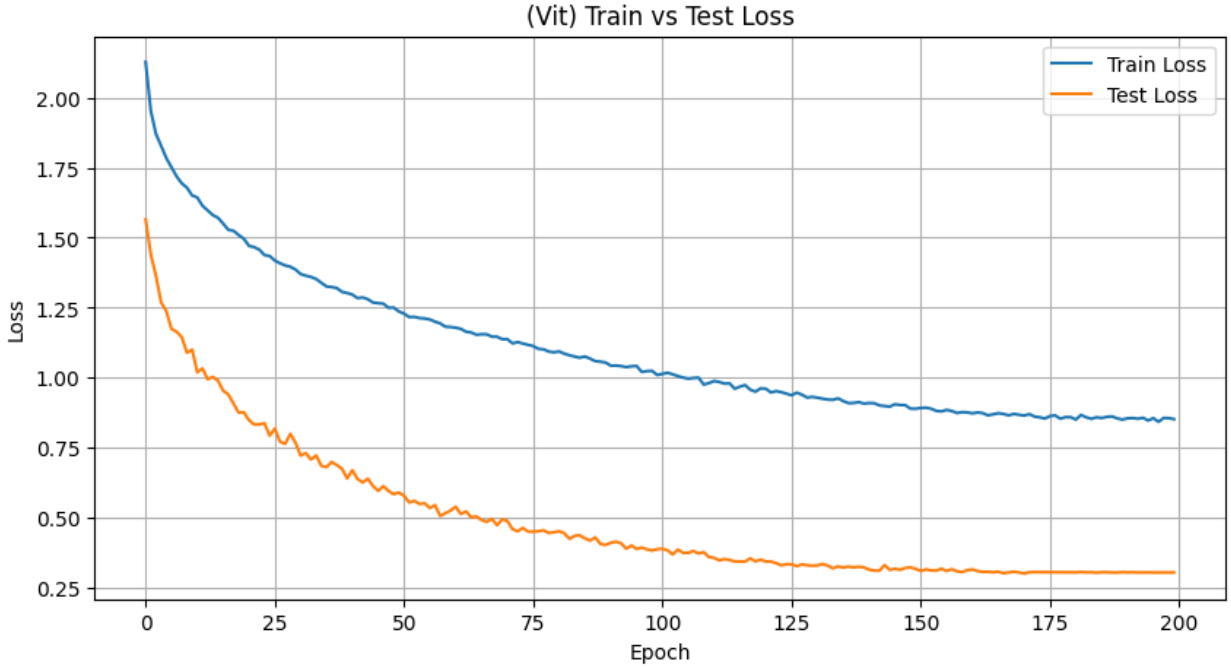
Conclusion:

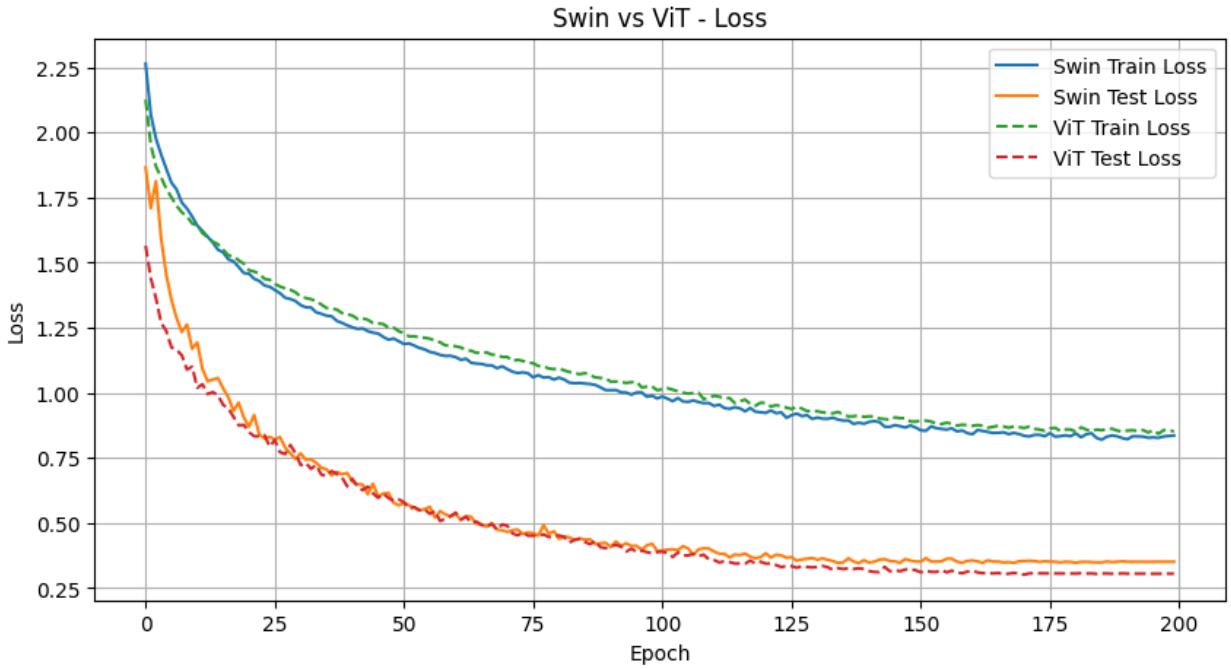
ViT is better than Swin Transformer on CIFAR-10 based on your current results. It generalizes slightly better and converges to a more optimal solution under the same conditions.

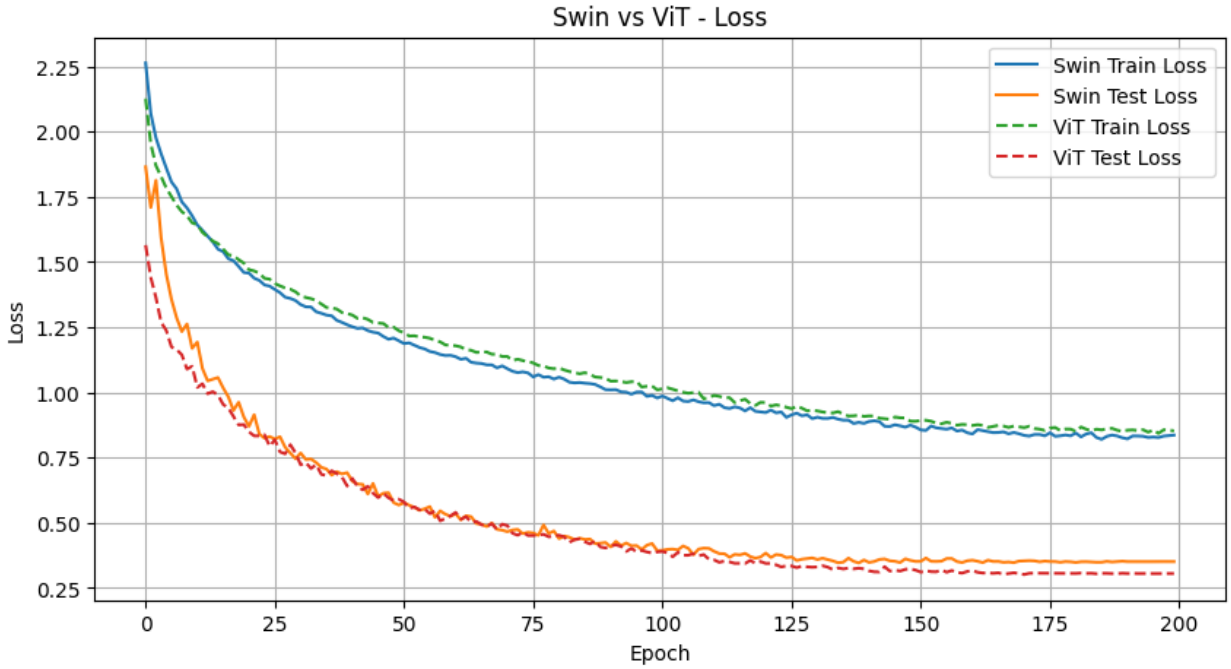

(Swin) Train vs Test Loss
Train Loss
Test Loss
Loss
Epoch
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0
25
50
75
100
125
150
175
200


(Swin) Train vs Test Loss
Train Loss
Test Loss
Loss
Epoch
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0
25
50
75
100
125
150
175
200


(Vit) Train vs Test Loss
Train Loss
Test Loss
Loss
Epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
25
50
75
100
125
150
175
200


(Vit) Train vs Test Loss
Train Loss
Test Loss
Loss
Epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
25
50
75
100
125
150
175
200


Swin vs ViT - Loss
Swin Train Loss
Swin Test Loss
ViT Train Loss
ViT Test Loss
Loss
Epoch
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
25
50
75
100
125
150
175
200


Swin vs ViT - Loss
Swin Train Loss
Swin Test Loss
ViT Train Loss
ViT Test Loss
Loss
Epoch
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
25
50
75
100
125
150
175
200