Modern Curriculum Learning in Computer Vision

ICARC Tutorial Session 2026

Asiri Gawesha
Faculty of Engineering
Sri Lanka Institute of Information Technology (SLIIT)

Why Curriculum Learning, and Why for Computer Vision?

Deep vision models are:

- **Large and expensive** to train (ImageNet-scale backbones, 3D segmentation, multimodal LLMs)
- Increasingly trained on **heterogeneous, noisy, or multimodal data**

Curriculum learning (CL):

- Structures training from **easy→hard** or **better-organized** data/targets over time
- Can improve:
  - **Efficiency** (wall-time, FLOPs, data usage)
  - **Accuracy / robustness**
  - **Stability** (variance, pseudo-label noise, task balance)

Purpose of this talk:

- Present 4 **representative curriculum methods** that illustrate different design philosophies and use-cases

Roadmap

Background & selection criteria

Four case studies:

- EfficientTrain – Intra-sample frequency & augmentation curriculum
- EfficientTrain++
- Pruning-Guided Curriculum Learning
- PGPS – Patch-size curriculum for 3D segmentation

EfficientTrain – Exploring Generalized Curriculum Learning for Training Visual Backbones

Why Rethink Curriculum Learning?

Traditional Sample-wise CL assumes:

•Samples can be ranked as easy → hard

•Train on easy samples first

But this has two major issues:

High Computational Cost

•Requires teacher networks or difficulty estimation

•Additional forward passes

•Dynamic scoring overhead

Ambiguous Sample Difficulty

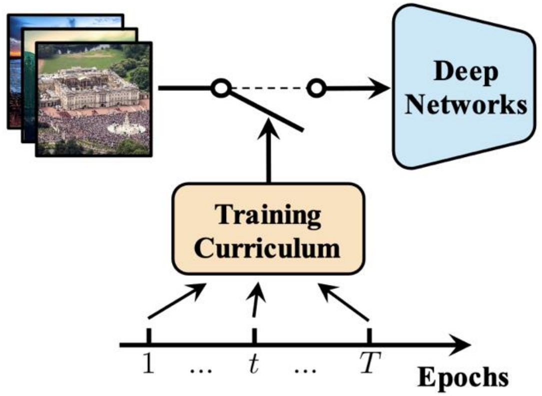
•“Hard” samples may contain critical discriminative features

•Some works show hard-to-easy can outperform easy-to-hard

•Difficulty ≠ usefulness

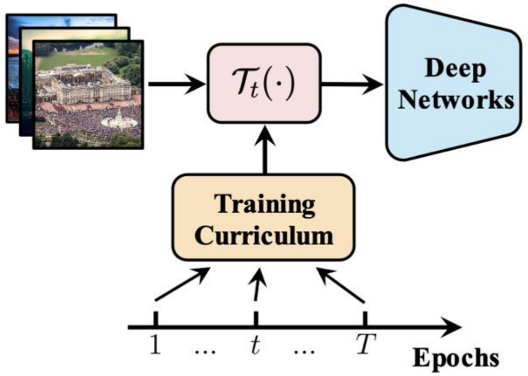
EfficientTrain asks:

What if difficulty is not at the sample level?



(a) Sample-wise CL (existing works)

Discrete-selection: ‘selecting easier-to-harder samples’



(b) Generalized CL (ours)

Soft-selection: ‘uncovering progressively more difficult patterns’

Core Hypothesis

What they did:

•Train a model normally on full-resolution images.

•At intermediate checkpoints, evaluate it on:

•Full images

•Low-pass filtered images (different bandwidths)

What they observed:

•Early checkpoints perform surprisingly well on **low-pass filtered vali**

•Performance drops much more on high-frequency-only images.

Generalized Curriculum Learning

Instead of ranking samples, they propose:

Every sample contains both easy-to-learn and hard-to-learn patterns.

Within a single image:

•Low-frequency components → easier

•High-frequency components → harder

Thus curriculum should be:

•Continuous

•Feature-level

•Transformation-based

Not sample selection.

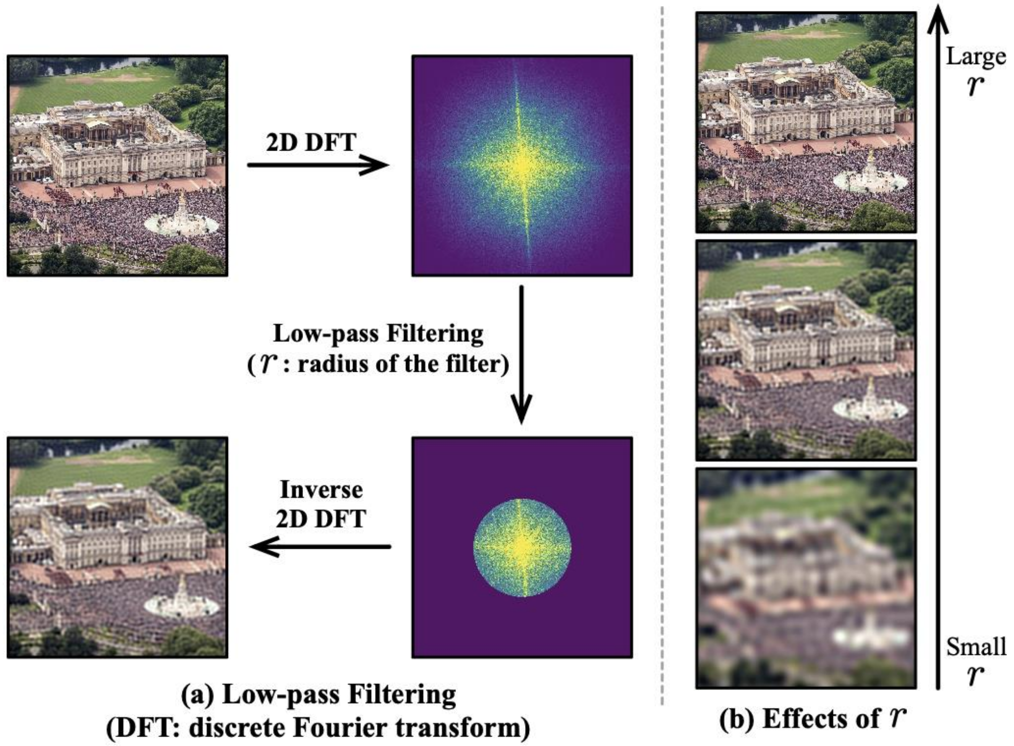


Figure 2: **Low-pass filtering.** Following [55], we adopt a circular filter.

Low pass filtering does not completely remove high frequencies

Method 1: Frequency-Domain Curriculum

Low-Frequency Cropping (DFT-based)

Instead of low pass filtering:
1. Convert image to Fourier domain
2. Crop central B × B region (low frequencies)
3. Inverse transform to pixel space

What does this means? how fast pixel values change in space

Low Frequencies = Slow Changes
Examples: Blue sky gradient, Large object shapes, Background color transitions
Mathematically: Pixel intensities change slowly over space.
Visually: Smooth regions.

High Frequencies = Rapid Changes
Examples: Edges ,Fur texture, Grass blades, Wrinkles, Noise
Mathematically: Pixel intensities change rapidly over small distances.
Visually: Sharp details.

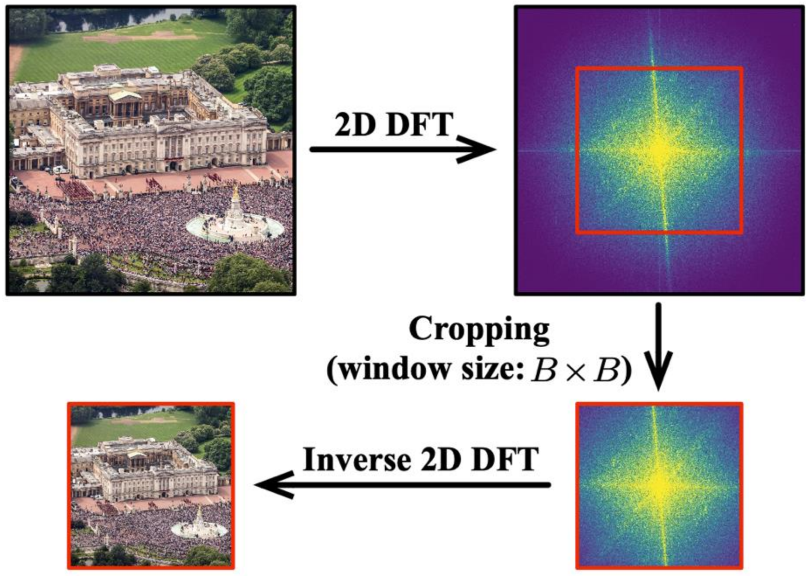


Figure 5: **Low-frequency cropping in the frequency domain** (B^2: bandwidth).

```
+--------------------------------+
| H  H  H  H  H  H  H  H |
| H  H  H  H  H  H  H  H |
| H  H  M  M  M  M  H  H |
| H  H  M  L  L  M  H  H |
| H  H  M  L  L  M  H  H |
| H  H  M  M  M  M  H  H |
| H  H  H  H  H  H  H  H |
| H  H  H  H  H  H  H  H |
+--------------------------------+
```

Method 2: Spatial-Domain Curriculum

Dynamic Data Augmentation
Observation:
•Unaugmented images are easier
•Heavy augmentation increases difficulty

Implementation:
•Use RandAug – Rotate, Flip, Change brightness, Shear, Cutout
•Linearly increase magnitude from 0 → 9 over training

This creates:
Weak → Strong augmentation curriculum

They combine both:
•Frequency bandwidth scheduling
•Augmentation strength scheduling
They model bandwidth as:

$$B = f(epoch)$$

Instead of fixed steps, they approximate continuous scheduling.

Methodology

```
+---------+
|   L     |
|  L L    |
+---------+
```

•Only smooth structure
•No fine details

Frequency Curriculum

•DFT → Crop center B×B (low frequencies)
•Gradually increase B
•Full resolution at end

Augmentation Curriculum

•RandAug magnitude: 0 → 9
•Weak → Strong augmentation

```
+--------------+
|   M M M      |
|  M L L M     |
|   M M M      |
+--------------+
```

•Shape boundaries
•Medium detail

Greedy Search algorithm to decide the scheduling

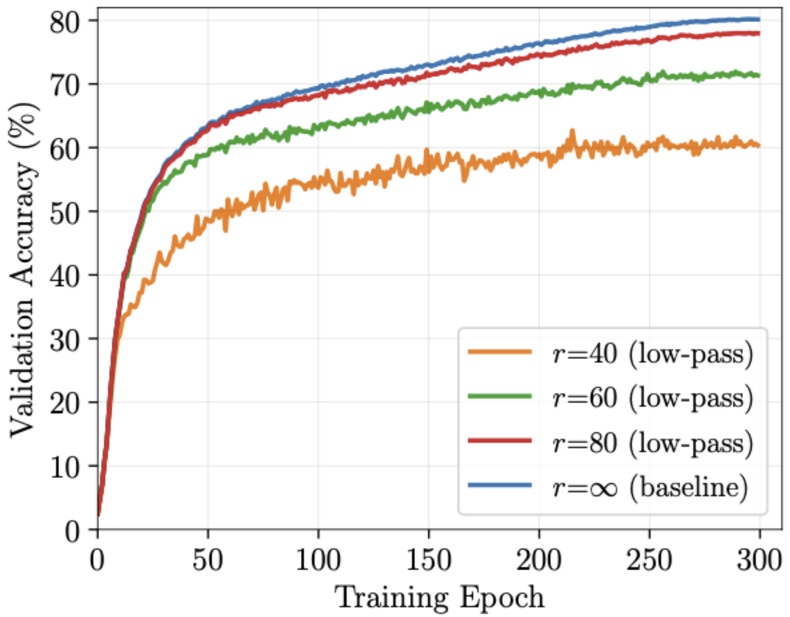
Epochs	Low-frequency Cropping	RandAug
1^{st} – 180^{th}	$B = 160$	$m = 0 \to 9$ Increase linearly.
181^{th} – 240^{th}	$B = 192$	
241^{th} – 300^{th}	$B = 224$	

```
+--------------------------------+
| H  H  H  H  H  H  H  H |
| H  H  H  H  H  H  H  H |
| H  H  M  M  M  M  H  H |
| H  H  M  L  L  M  H  H |
| H  H  M  L  L  M  H  H |
| H  H  M  M  M  M  H  H |
| H  H  H  H  H  H  H  H |
| H  H  H  H  H  H  H  H |
+--------------------------------+
```

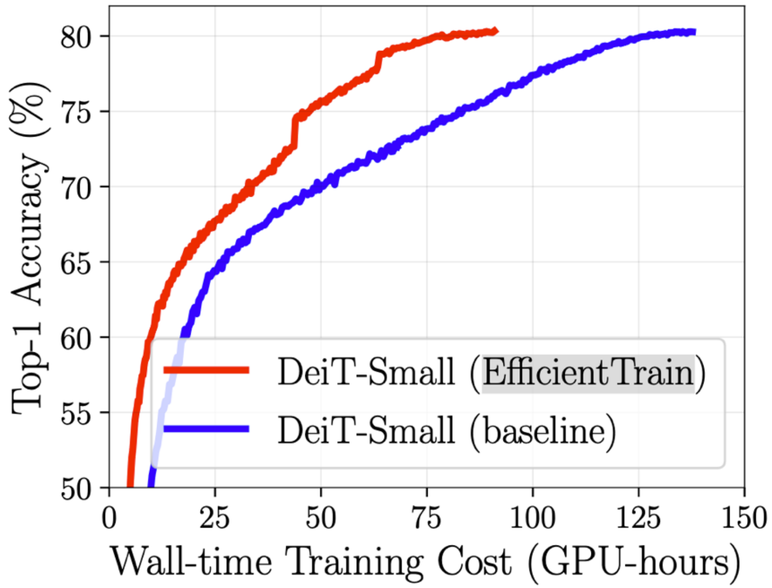
Results

EfficientTrain works across:
•CNNS
•Vision Transformers
No architecture-specific design
Dataset - ImageNet-1 K

Model	Baseline	EfficientTrain	Speedup
ResNet-50	78.8%	79.4%	1.44×
ConvNeXt-T	82.1%	82.2%	1.49×
Swin-B	83.4%	83.6%	1.50×
CSWin-B	84.3%	84.3%	1.56×



Train: Original Images *(Low+High Frequency)*;
Val. : Low-pass Filtered Images;

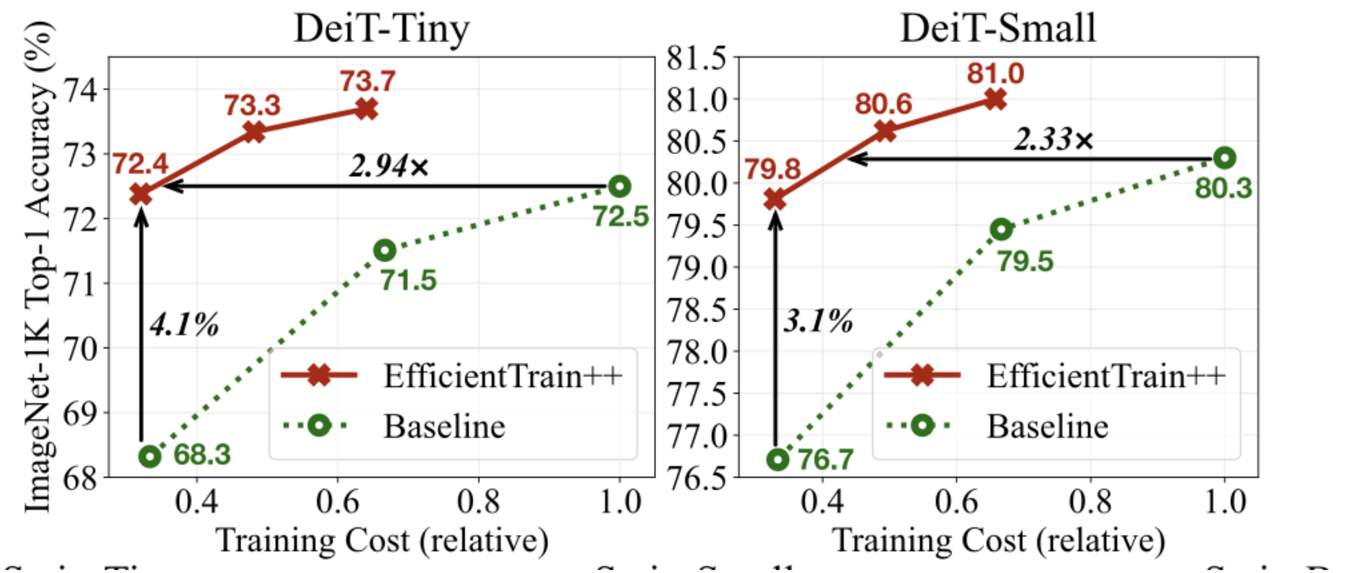


EfficientTrain++: Generalized Curriculum Learning for Efficient Visual Backbone Training (2024)

EfficientTrain (ICLR 2023)
•Frequency + augmentation curriculum
•Greedy schedule search
•~1.5× speedup

Limitations
•Schedule search expensive
•Stage transitions abrupt
•I/O bottlenecks
•Not fully optimized for large-scale

EfficientTrain++ Goal
Make curriculum scalable and system-efficient
Used a sequential searching algorithm instead of greedy search



Greedy search VS Sequential search

Original approach:
1.Define candidate bandwidth values
2.Divide training into stages
3.Run multiple experiments
4.Pick best performing schedule

•At each stage, they pick the locally best next bandwidth.
•They don't globally optimize the entire schedule.
•They make a locally optimal decision step-by-step.
Good for research prototype — not ideal for large-scale production.

Trial 1 → Evaluate
Trial 2 → Evaluate
Trial 3 → Evaluate
Select Best

Start with smallest safe bandwidth B1
- Train for short warm-up period

Evaluate validation performance
- Measure improvement vs baseline trend

Decide next bandwidth Bk+1Bk+1
- Increase if performance saturates
- Keep small if still improving

Update compute budget tracker
- Ensure total FLOPs remain within constraint

Repeat until full bandwidth reached

Single Run
Sequential Adjustment

Results

Datasets - ImageNet-22K, Large ViT backbones
Additional optmizations
Optimized Low-Frequency Processing Pipeline
Increase batch size in early stage.
From ~1.5× → up to ~3× wall-time speedup

- **Up to ~3× wall-time speedup**
- Same or slightly better Top-1 accuracy
- Works across:
- ResNet
- ConvNeXt
- Swin
- DeiT
- CSWin
- CAFormer

Version	Speedup
EfficientTrain	~1.5×
EfficientTrain++	Up to ~3×

Pruning-Guided Curriculum Learning for Semi-Supervised Semantic Segmentation (2023)

What is semantic segmentation?

- Pixel-wise classification of an image.
- Instead of predicting one label per image, predict one label per pixel

What is semi supervised learning?

•Small labeled dataset

•Large unlabeled dataset

Goal:

Use unlabeled images to improve performance.

Standard SSL pipeline:

Train teacher on labeled data.

Use teacher to generate pseudo-labels on unlabeled images.

Train student on both labeled and pseudo-labeled data.

This is called:

Mean Teacher framework.

What are they trying to do?

Improve pseudo-label quality in semi-supervised semantic segmentation.

Pseudo-labels can be:
•Noisy
•Overconfident
•Especially unreliable early in training

So the paper proposes:
Use pruning to estimate reliability and apply curriculum learning to gradually include harder pseudo-labels.

Network pruning disproportionately hurts poorly learned samples.

So they hypothesize:
If pruning strongly changes a pixel's feature,
→ that pixel is not well learned yet.
That means:
•Its confidence score is unreliable.

Input Confidence Ours

They introduce a **third network branch**:
•Student (trained normally)
•Teacher (EMA of student)
•**Pruned Teacher**

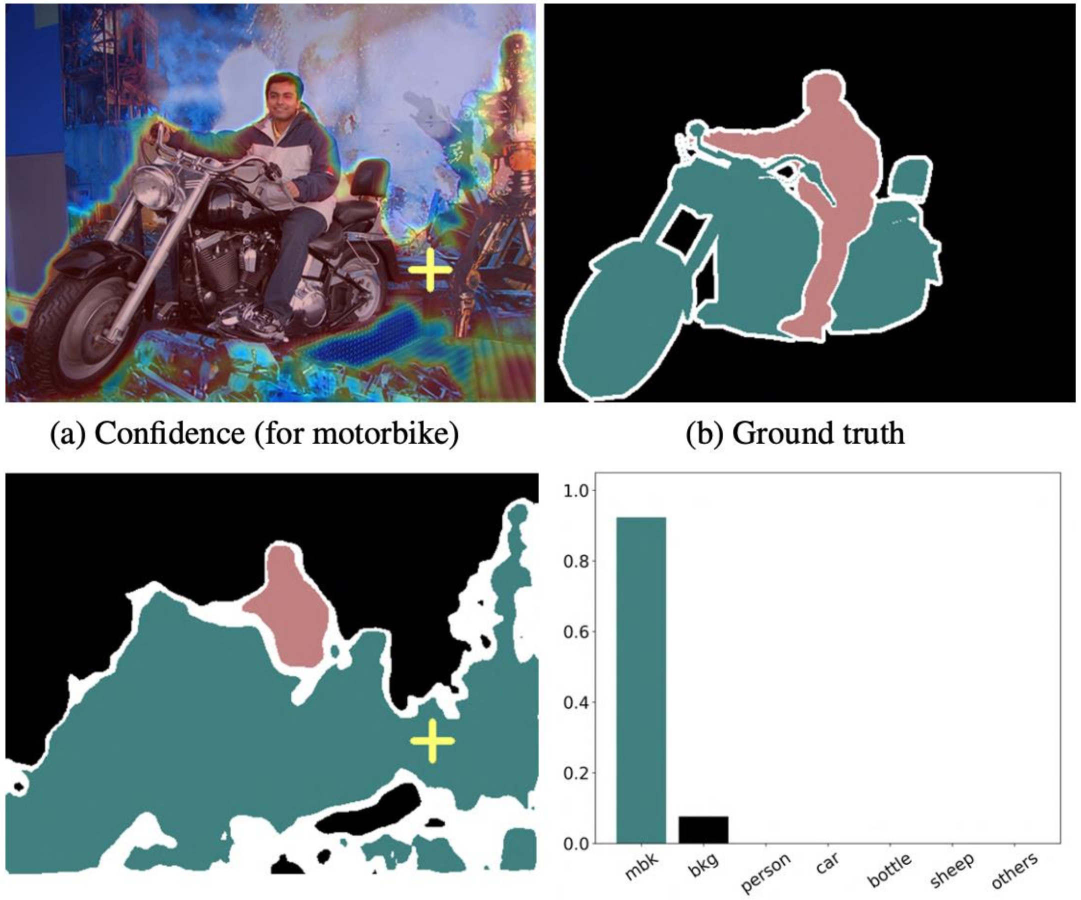
Methodology

Apply magnitude-based pruning to teacher encoder.
Mask updated once per epoch.
For each pixel:
•Extract embedding from teacher:
•Extract embedding from pruned teacher:
•Compute cosine similarity:

$$d(\tilde{z}_i, \tilde{z}_i^p)$$

If similarity is low:
→ pruning changed representation a lot
→ sample not well learned
→ unreliable pseudo-label.

final score = softmax × pruning stability.
If pruning hurts prediction:
confidence is reduced.



(a) Confidence (for motorbike) (b) Ground truth
(c) Pseudo-labels after filtering (d) Softmax probability (yellow cross)

How CL comes into play?

But strong pruning suppresses hard samples too much.

So they introduce **self-paced curriculum**:

$$\vartheta_t = \vartheta_{max} - (\vartheta_{max} - \vartheta_{min}) \left(\frac{t}{t_{max}} \right)^{\varsigma}$$

•Early training → strong pruning influence → Only easy pseudo-labels used
•Later training → pruning influence decreases → Hard samples gradually included
This is **explicit easy → hard scheduling**.

Traditional CL:
Sample-level (whole image)
PGCL:
Pixel-level curriculum
Difficulty is computed per pixel.
This is fine-grained curriculum learning.

Experimental Results

Datasets
•PASCAL VOC 2012
•Cityscapes

Backbone
•DeepLabV3+
•ResNet-50
•ResNet-101

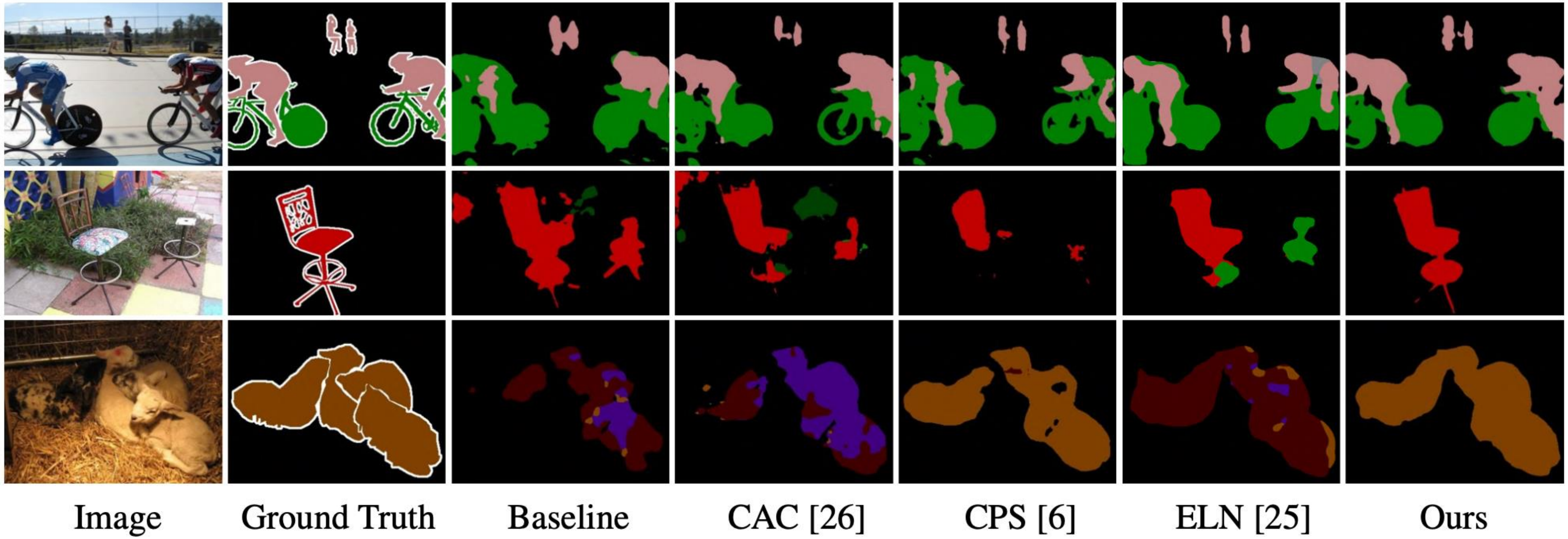
Method	Backbone	mIoU
Baseline (Mean Teacher)	ResNet-50	68.2
PGCL (Ours)	ResNet-50	**75.2**
Baseline	ResNet-101	71.5
PGCL (Ours)	ResNet-101	**76.8**

PASCAL VOC (1/8 labeled split)

Method	Backbone	mIoU
Baseline	ResNet-50	~69
PGCL (Ours)	ResNet-50	**~73+**

Cityscapes (1/8 labeled split)

Semantic segmentation results



Progressive Growing of Patch Size: Curriculum Learning for Accelerated and Improved Medical Image Segmentation

Volumes are very large (memory heavy)

- Foreground objects are small compared to background
- Severe class imbalance
- Full volume cannot fit in GPU memory

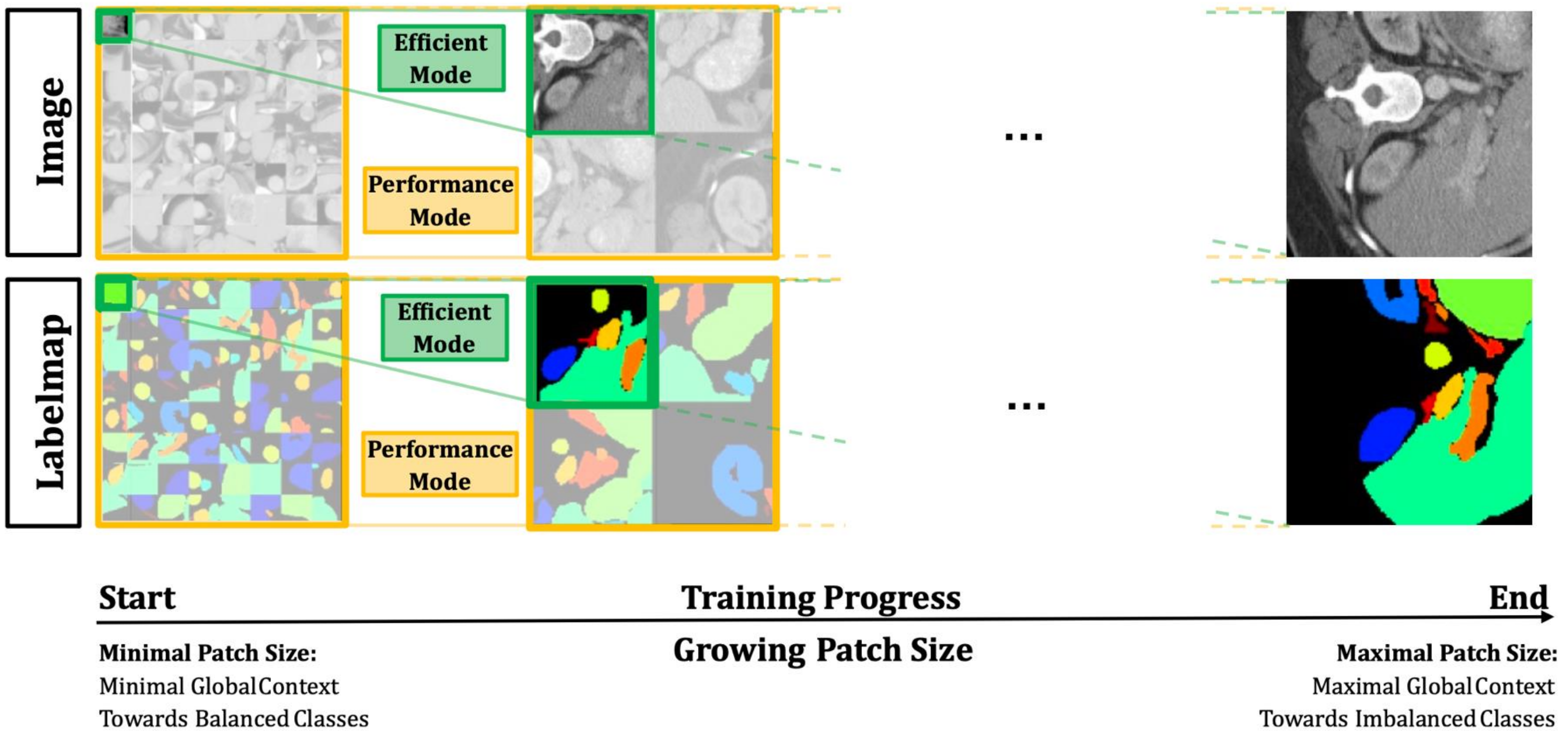
Instead of feeding entire 3D volume:
They crop smaller 3D patches:
Patch⊂Volume
This reduces memory usage.

They evaluate on many datasets such as:

- MSD (Medical Segmentation Decathlon)
- KiTS23 (Kidney Tumor Segmentation)
- AMOS22
- BTCV
- TotalSegmentatorV2
- ToothFairy2

These include segmentation of:

- Liver
- Kidney
- Tumors
- Spleen
- Pancreas
- Multi-organ structures

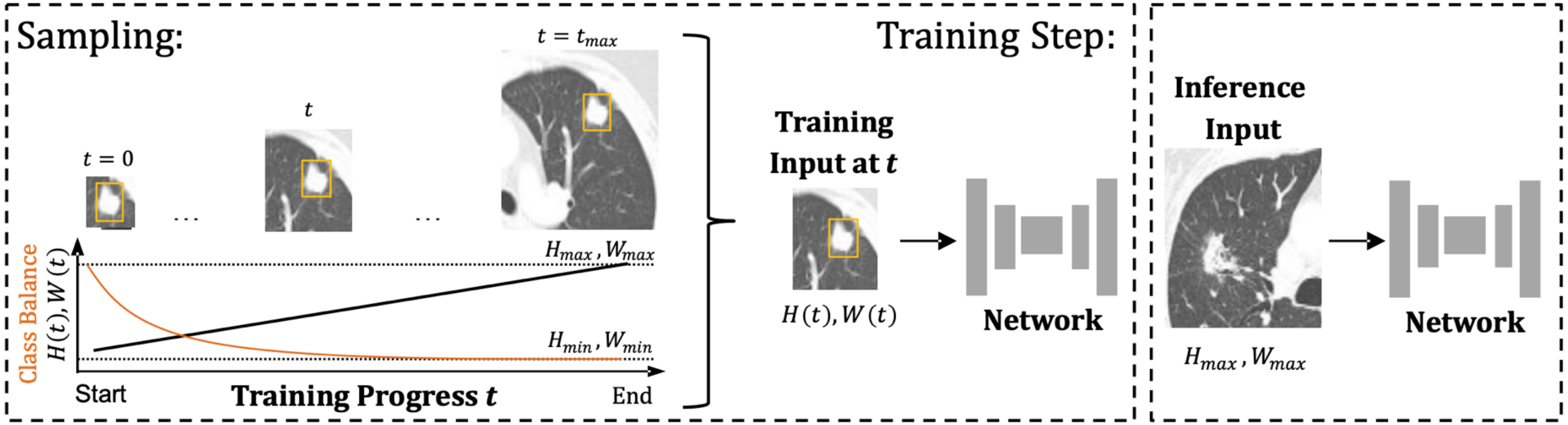


- Each Voxel - Intensity value
- Tissue density
- MRI signal value

Progressive Growing of Patch Size: Curriculum Learning for Accelerated and Improved Medical Image Segmentation

Progressive Growing of Patch Size (PGPS) for 3D Med Seg
•Patch-based training → limited context + class imbalance
•Usual baseline: constant *max* patch size (nnU-Net)
Idea: patch size as curriculum
In 3D segmentation:
Large patches:
•Contain lots of background voxels
•More spatial context to model
Small patches:
•More foreground concentration
•Simpler spatial modeling
So:
Small patch → easier
Large patch → harder
Additionally increase the batch size
Patch size acts as difficulty proxy.

Saved ~**44%** training time, ~**33%** FLOPs



Backbone	CPS	PGPS-Eff	PGPS-Perf
UNet (nnU-Net) [2]	<u>83.37</u>	83.05	**83.81**
UNETR [5]	<u>71.20</u>	0.00	**72.76**
SwinUNETR [6]	71.62	<u>71.98</u>	**75.39**

SLIIT UNI THE KNOWLEDGE UNIVERSITY
Utrecht University
ICARC
BRAIN LABS

Thank you!

Any questions?