University of Zurich[UZH]

UZH Group Meeting

Zurich, 5th November 2024

Data Augmentation Studies for Binary Classification

Charlie Rohrbach[1], Vagelis Gkougkousis[1,2]

[1]University of Zurich
[2]CERN

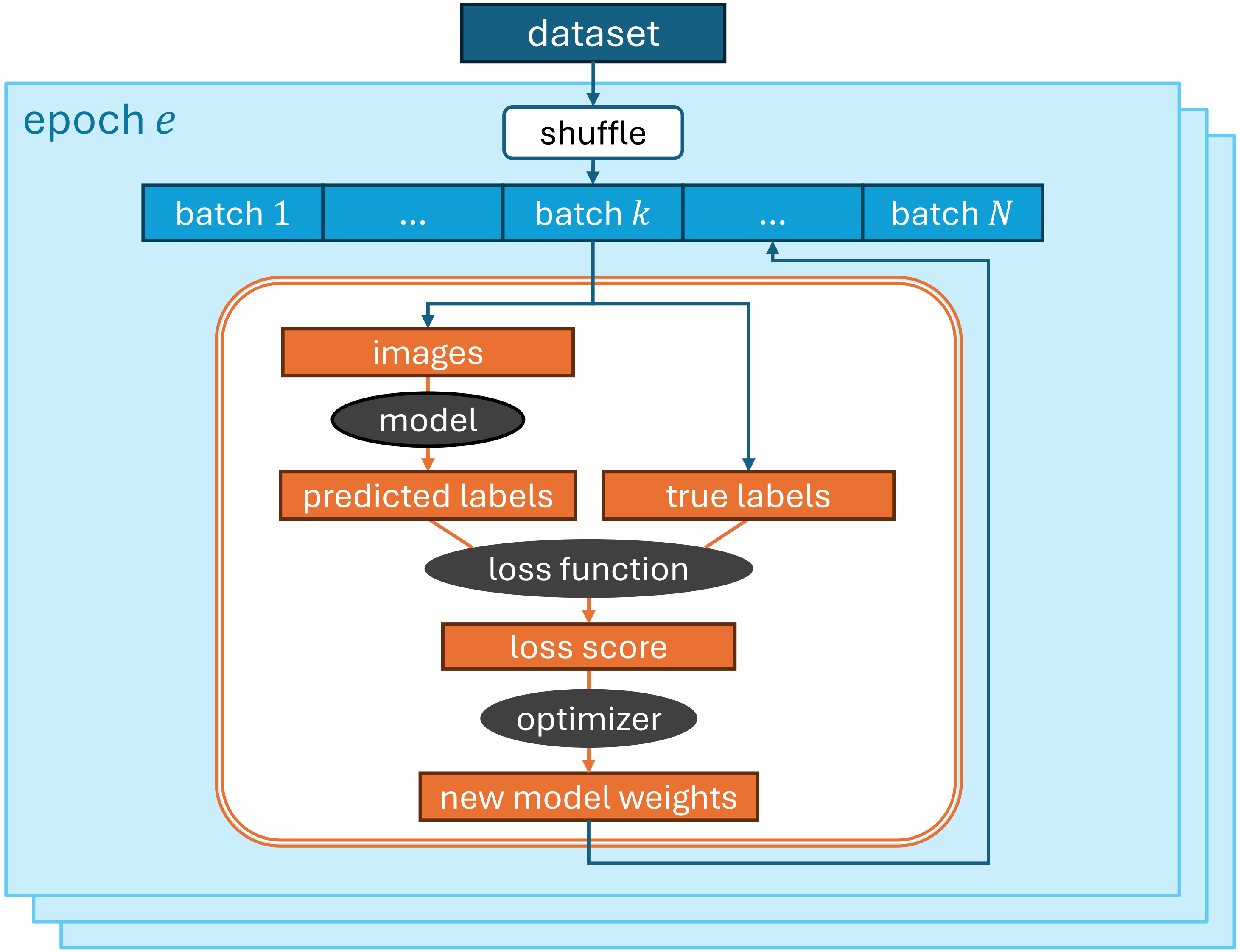
Base model structure

Layer 0	Input	• Shape: (128, 128, 3)
Layer 1, 5, 9	Convolution 2D	• Kernel size: 3x3 • Activation: ReLU
Layer 2, 6, 10	Batch Normalisation	
Layer 3, 7, 11	Max Pooling 2D	• Size: 2x2
Layer 4, 8, 12	Dropout	• Rate: 0.25
Layer 13	Flatten	
Layer 14	Dense	• Neurons: 512 • Activation: ReLU
Layer 15	Dropout	• Rate: 0.5
Layer 16	Dense	• Neurons: 1 • Activation: sigmoid

× 3

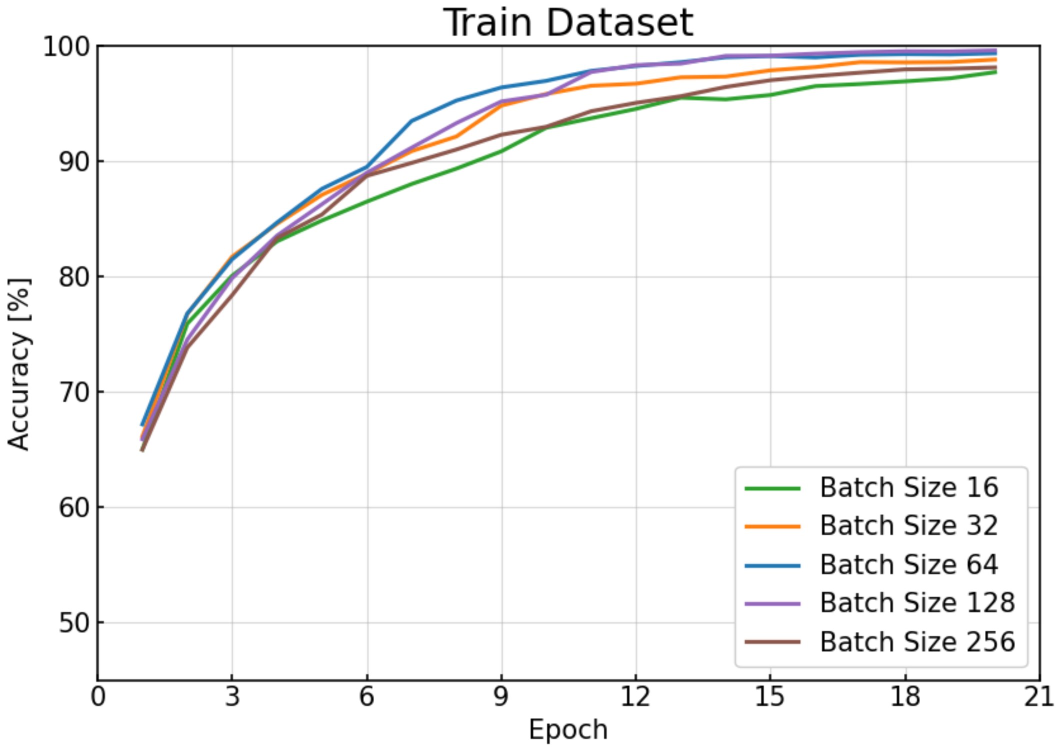
convolution filters: 32, 64, 128

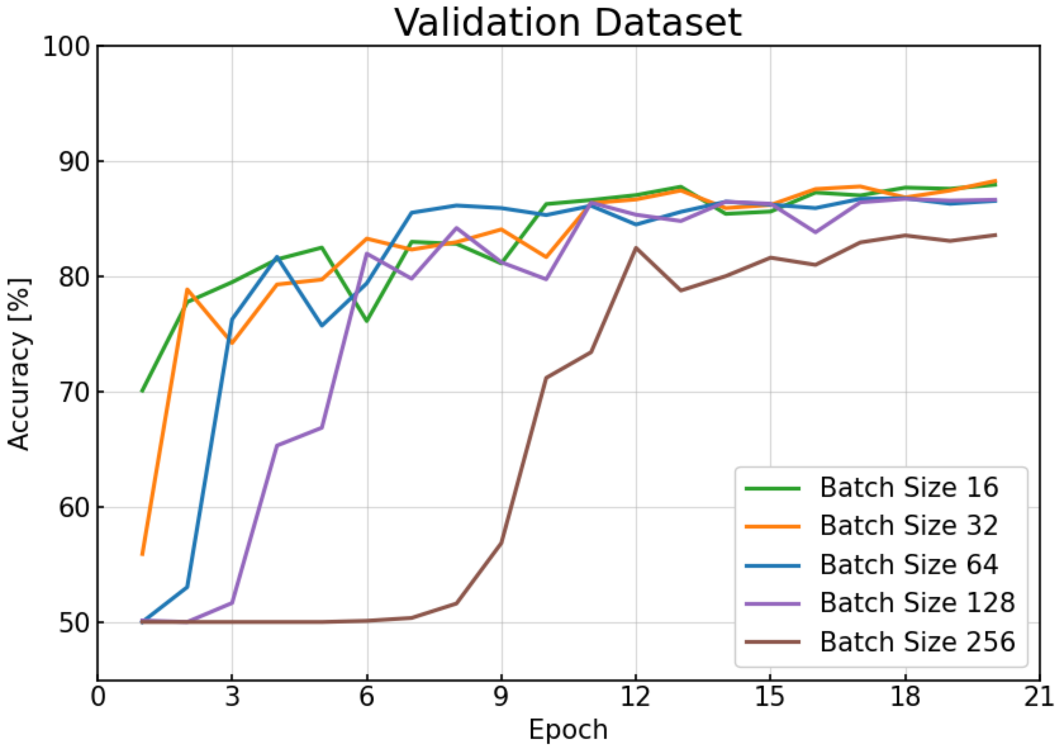
Training process



Base model

- High accuracy on training data, limited accuracy on validation data





Batch Size	16	32	64	128	256
Final Acc. [%]	97.7	98.8	99.3	99.6	98.1

Batch Size	16	32	64	128	256
Final Acc. [%]	87.9	88.3	86.5	86.7	83.6

Data augmentation

Wrap-around
Tracking Application

Input
- Shape: (128, 128, 3)

Augmentation

Random Zoom
- range: -20%, 20%
- fill mode: wrap

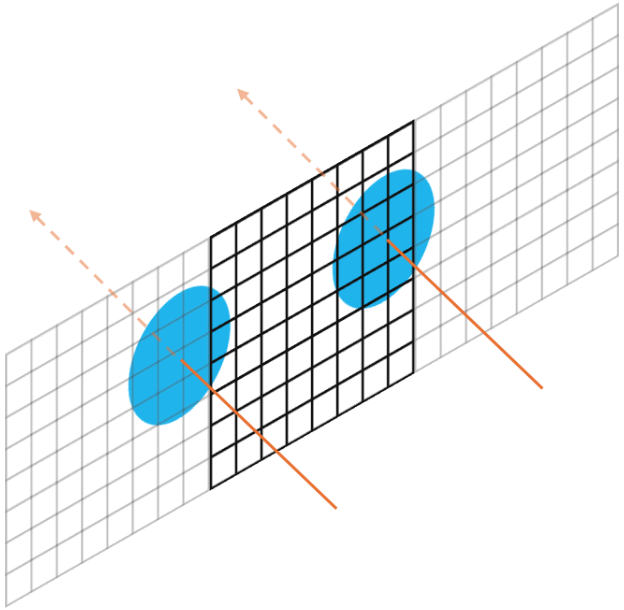
Random Flip
- horizontal
- probability: 50%

Random Translation
- range: -10%, 10% (height & width)
- fill mode: wrap

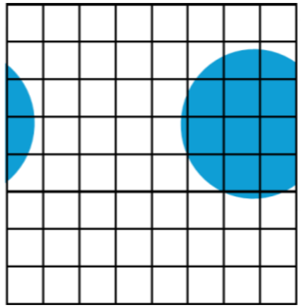
Base Model

Additional particle in adjacent chip creates signal that spills over

Full picture



Picture in single chip



Augmentation variations

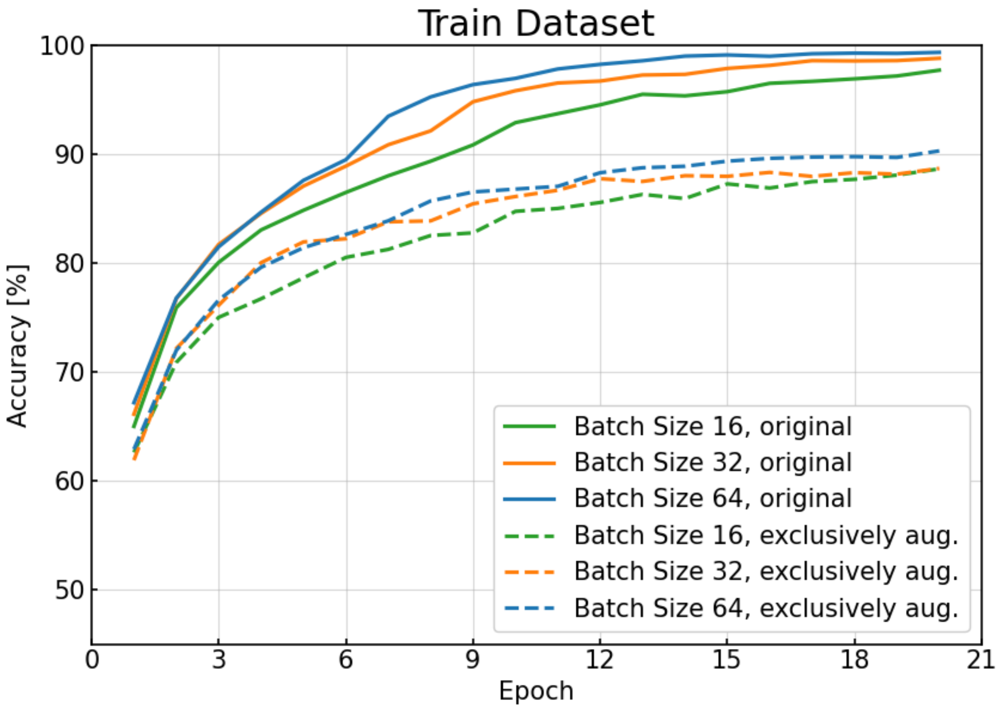
Exclusively augmented	Data mixing	Data doubling
Idea: • Apply augmentation to the full dataset	Idea: • Apply augmentation to a fraction of the dataset Use case: • Modify distribution of training data	Idea: • Use full original dataset • Combine with augmented copy of dataset Use case: • Increase volume of training data

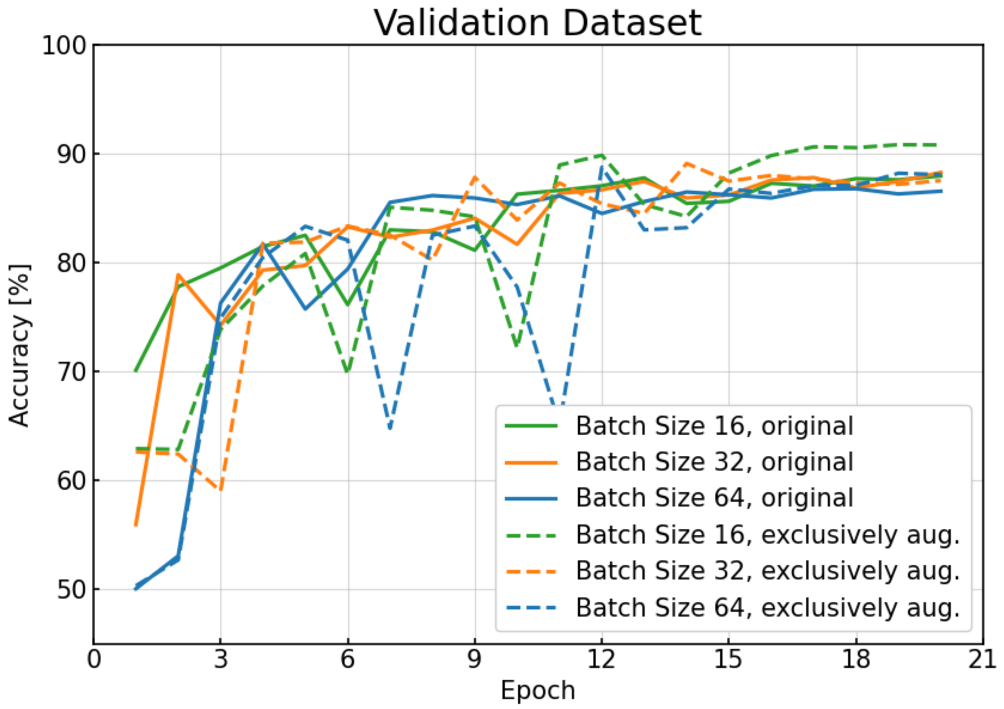
Augmentation variations

Exclusively augmented	Data mixing	Data doubling
Idea: • Apply augmentation to the full dataset	Idea: • Apply augmentation to a fraction of the dataset Use case: • Modify distribution of training data	Idea: • Use full original dataset • Combine with augmented copy of dataset Use case: • Increase volume of training data

Base model: Original vs. Exclusively augmented

- Augmentation is applied to training data, validation data remains the same for each model & epoch



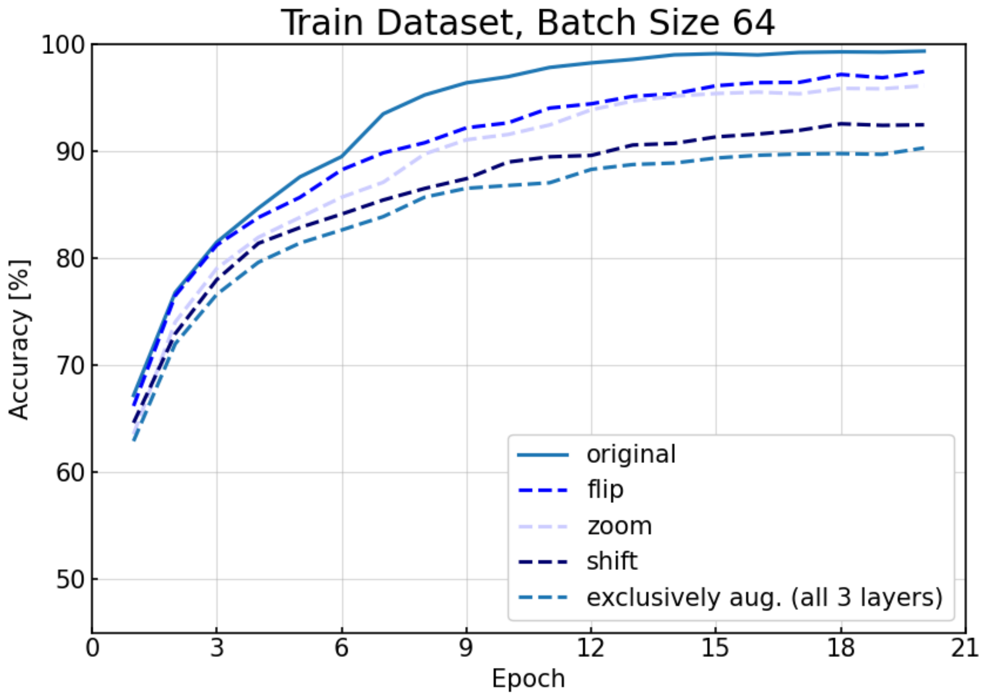


	Original			Exclusively augmented		
Batch Size	16	32	64	16	32	64
Final Acc. [%]	97.7	98.8	99.3	88.7	88.7	90.3

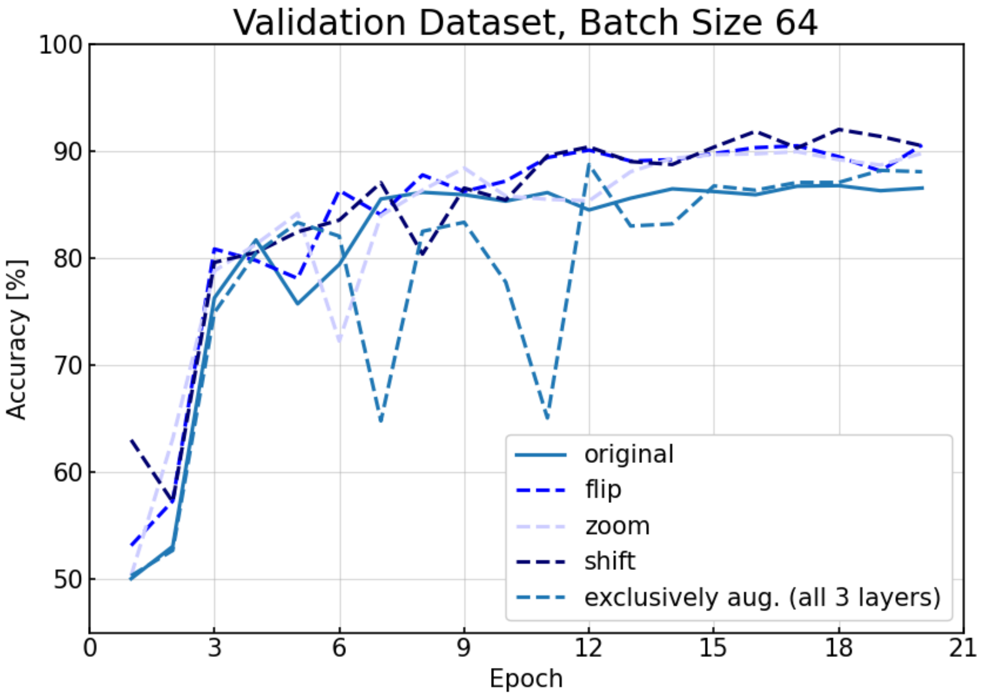
	Original			Exclusively augmented		
Batch Size	16	32	64	16	32	64
Final Acc. [%]	87.9	88.3	86.5	90.8	87.5	88.1

Augmentation: Layer breakdown

- Largest accuracy reduction from shift layer due to wrap-around
- Combination of all layers less accurate than any single layer



Model	Original	Flip	Zoom	Shift	Excl. aug.
Final Acc. [%]	99.3	97.4	96.1	92.5	90.3



Model	Original	Flip	Zoom	Shift	Excl. aug.
Final Acc. [%]	86.5	90.6	89.8	90.5	88.1

Augmentation variations

Exclusively augmented	Data mixing	Data doubling
Idea: • Apply augmentation to the full dataset	Idea: • Apply augmentation to a fraction of the dataset Use case: • Modify distribution of training data	Idea: • Use full original dataset • Combine with augmented copy of dataset Use case: • Increase volume of training data

Data mixing: Systematic

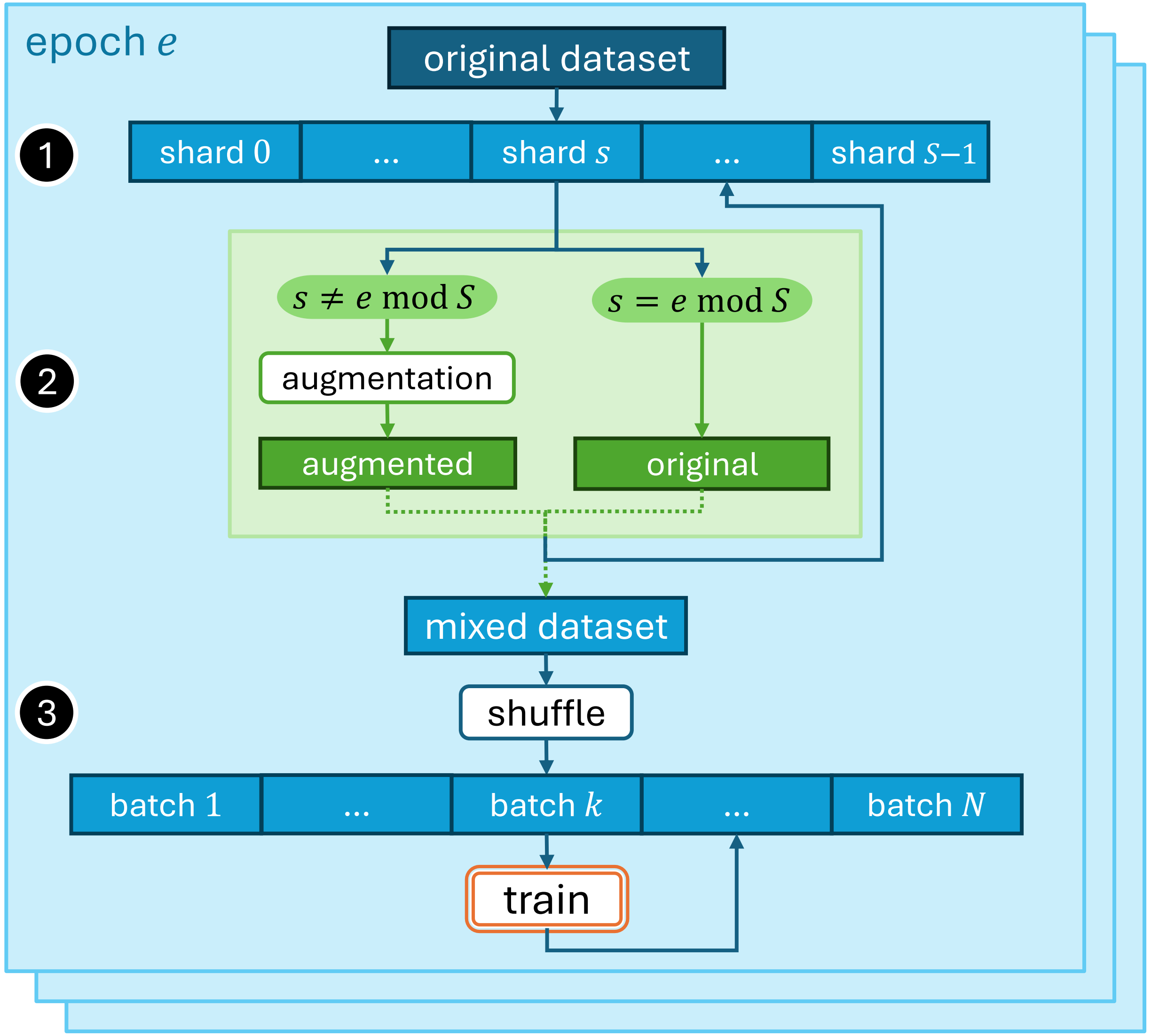
- Free parameter S: Total number of shards
  - Defines the fraction of original data as $1/S$

1) Divide the dataset into S equal-sized shards
2) In each epoch, augment each shard except 1
   - In the first epoch, skip the first shard
   - In each subsequent epoch, move to the next shard
   - After the last shard, start over at the beginning
3) Fully shuffle the dataset to get mixed batches

Example

- $S = 3$
- 34% original data
- 66% augmented data

shard 0	shard 1	shard 2



Data mixing: Systematic

- Free parameter S: Total number of shards
  - Defines the fraction of original data as $1/S$

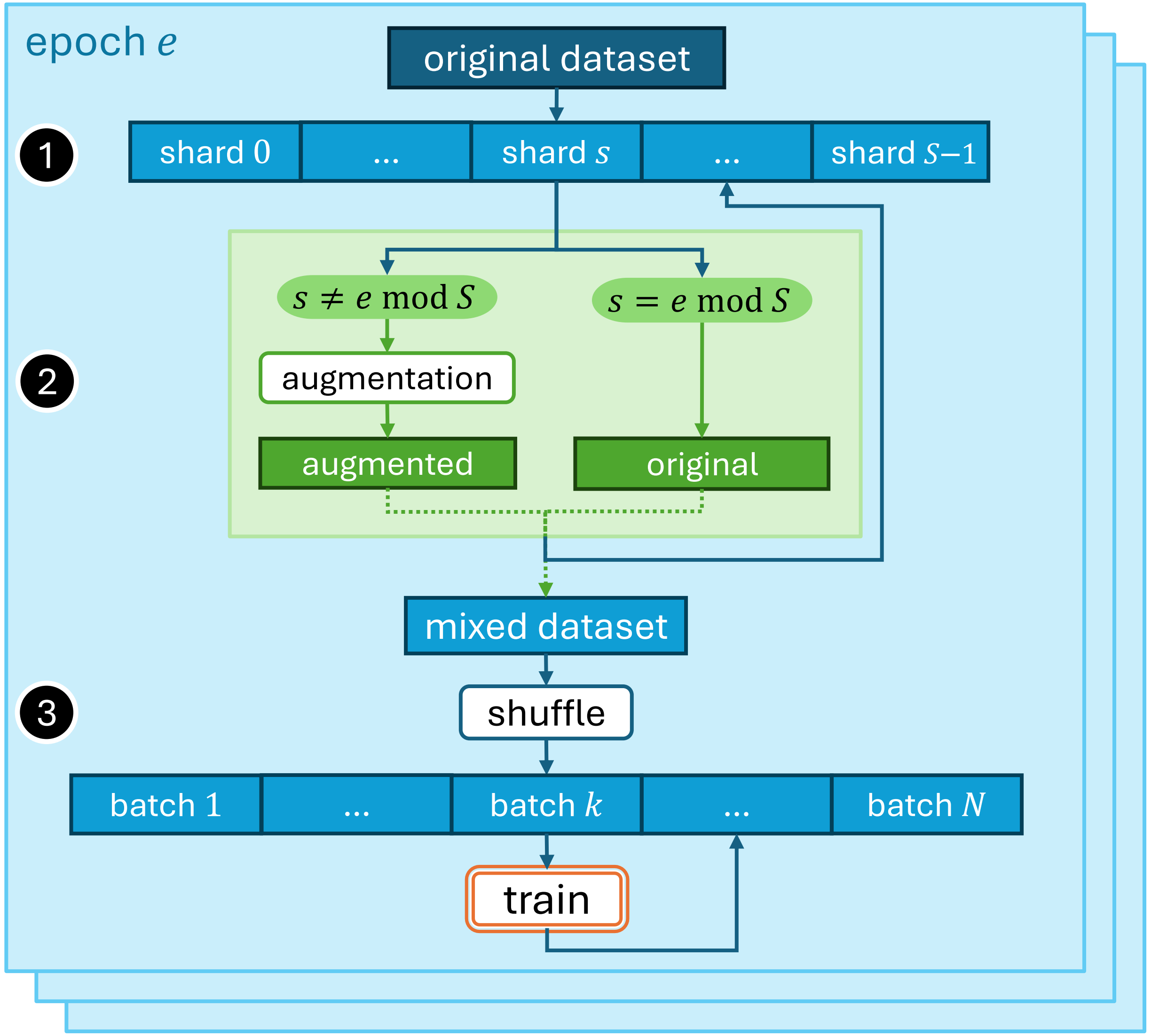
1) Divide the dataset into S equal-sized shards
2) In each epoch, augment each shard except 1
   - In the first epoch, skip the first shard
   - In each subsequent epoch, move to the next shard
   - After the last shard, start over at the beginning
3) Fully shuffle the dataset to get mixed batches

Example

- $S = 3$
- 34% original data
- 66% augmented data

$\boldsymbol{e = 0}$

shard 0	shard 1	shard 2



Data mixing: Systematic

- Free parameter S: Total number of shards
  - Defines the fraction of original data as $1/S$

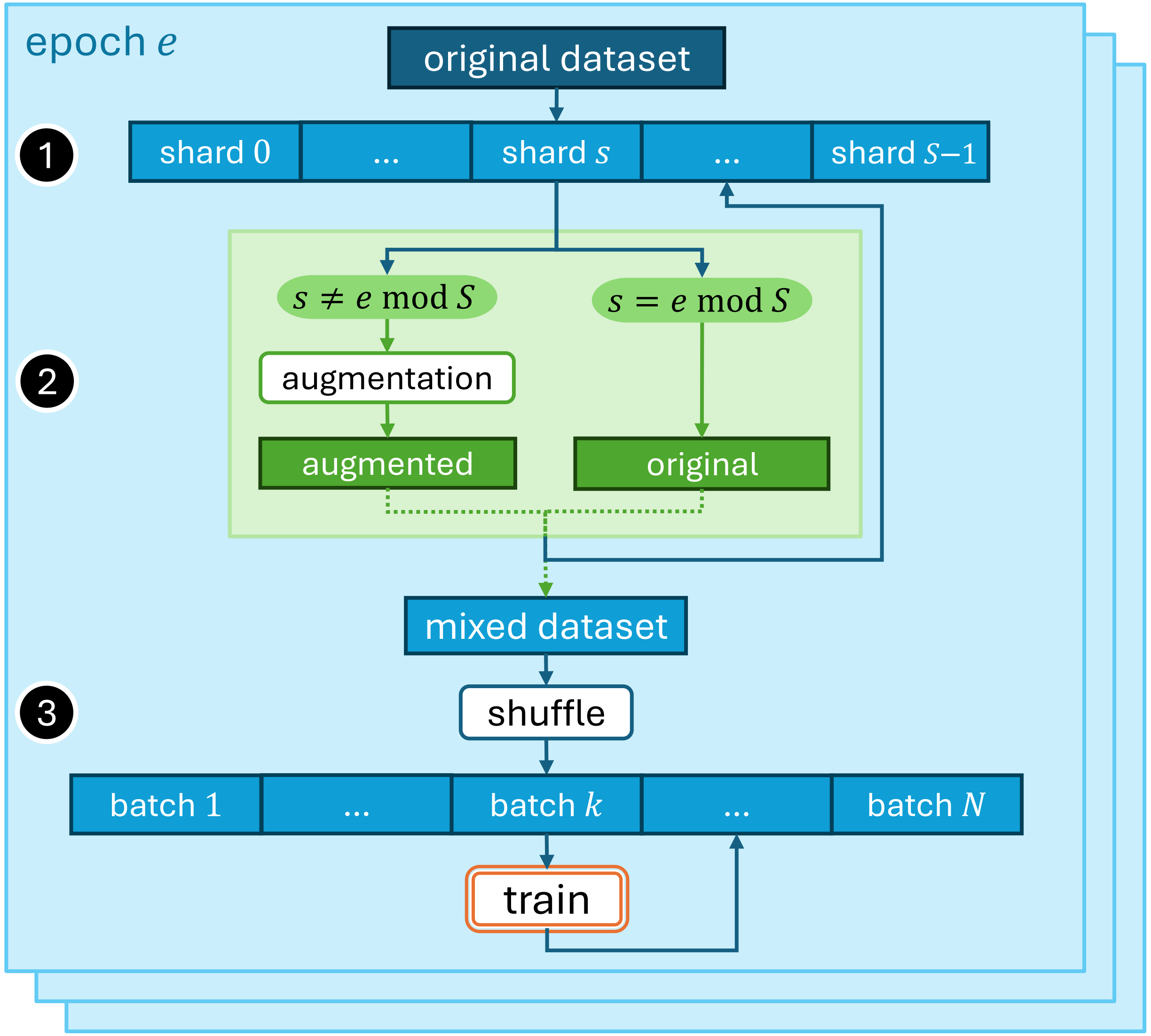
1) Divide the dataset into S equal-sized shards
2) In each epoch, augment each shard except 1
   - In the first epoch, skip the first shard
   - In each subsequent epoch, move to the next shard
   - After the last shard, start over at the beginning
3) Fully shuffle the dataset to get mixed batches

Example

- $S = 3$
- 34% original data
- 66% augmented data

$e = 1$

shard 0	shard 1	shard 2



Data mixing: Systematic

- Free parameter S: Total number of shards
  - Defines the fraction of original data as $1/S$

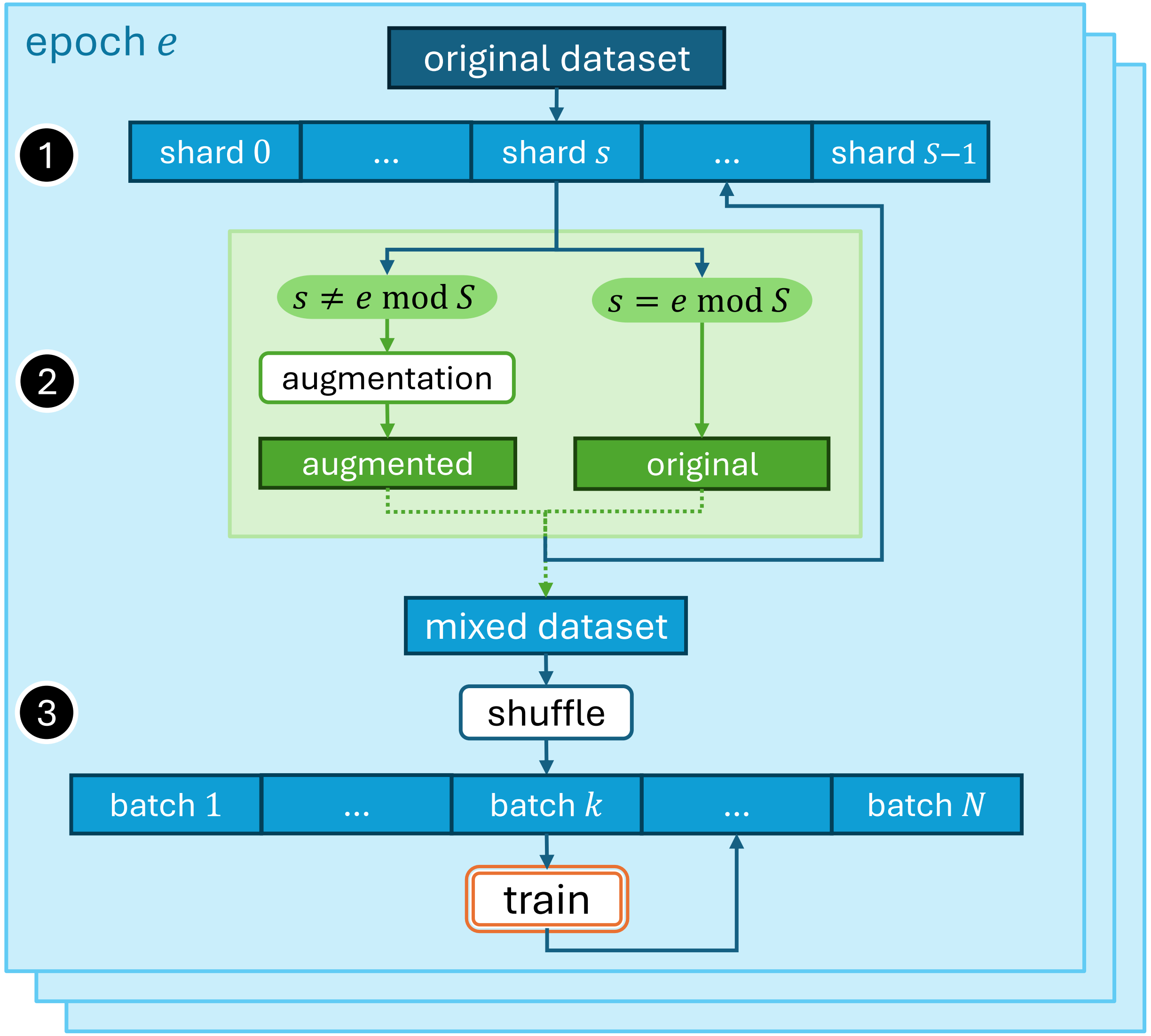
1) Divide the dataset into S equal-sized shards
2) In each epoch, augment each shard except 1
   - In the first epoch, skip the first shard
   - In each subsequent epoch, move to the next shard
   - After the last shard, start over at the beginning
3) Fully shuffle the dataset to get mixed batches

Example

- $S = 3$
- 34% original data
- 66% augmented data

$e = 2$

shard 0	shard 1	shard 2



Data mixing: Systematic

- Free parameter S: Total number of shards
  - Defines the fraction of original data as $1/S$

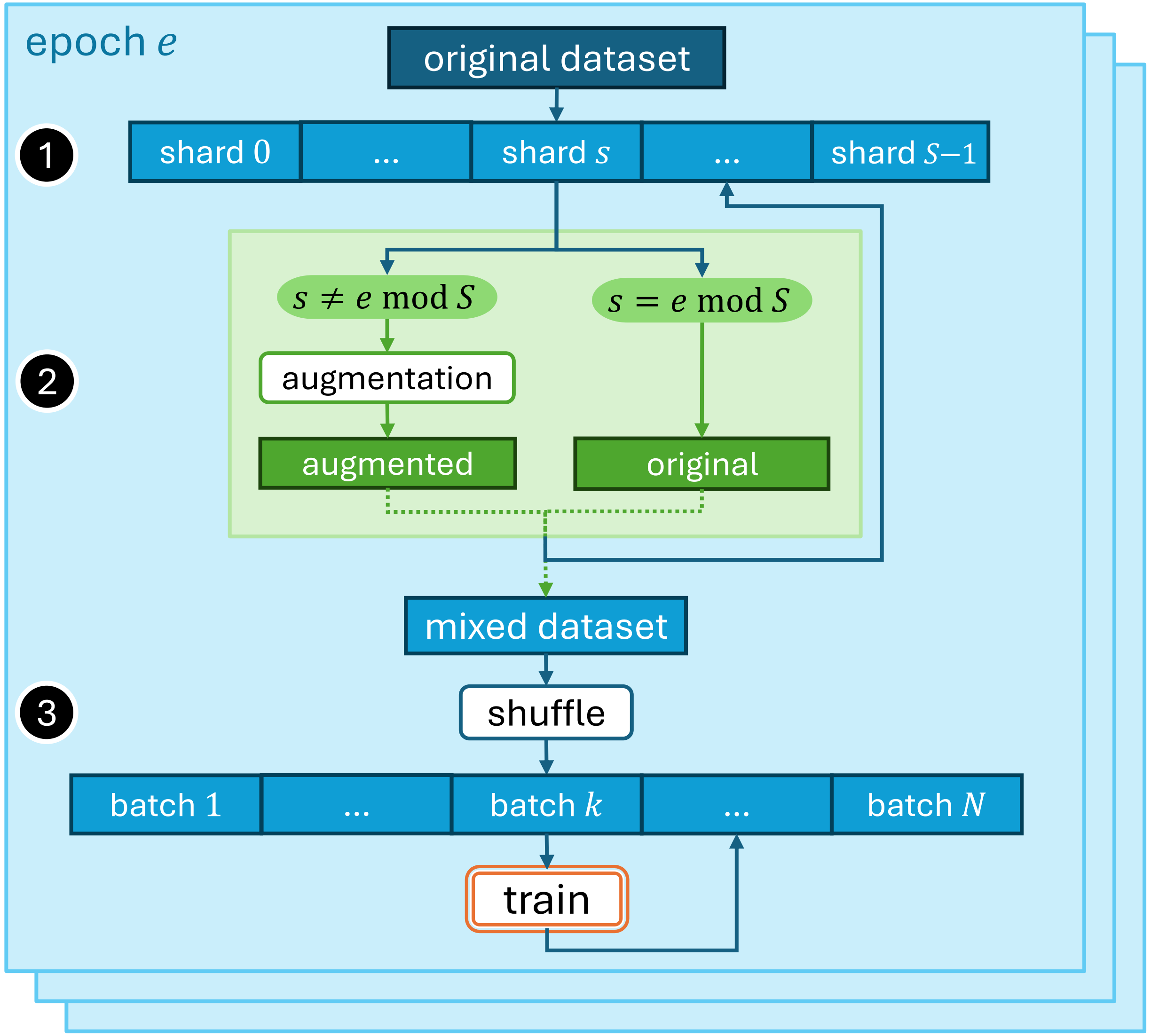
1) Divide the dataset into S equal-sized shards
2) In each epoch, augment each shard except 1
   - In the first epoch, skip the first shard
   - In each subsequent epoch, move to the next shard
   - After the last shard, start over at the beginning
3) Fully shuffle the dataset to get mixed batches

Example

- $S = 3$
- 34% original data
- 66% augmented data

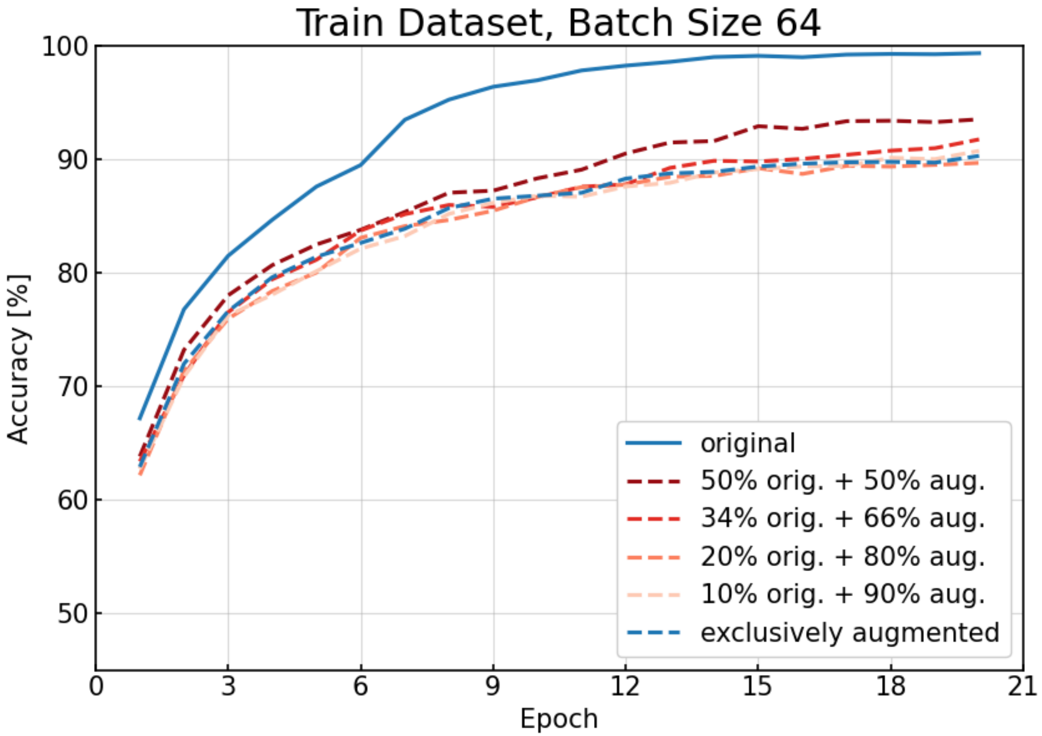
$e = 3$

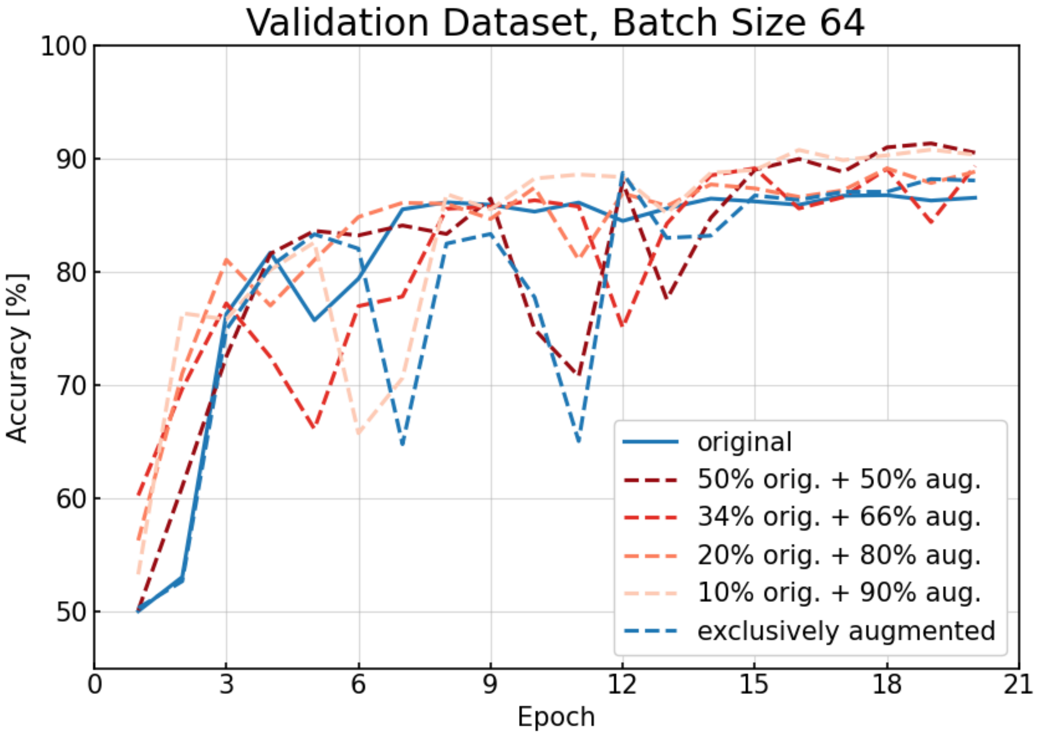
shard 0	shard 1	shard 2



Data mixing: Systematic

- Lower training accuracy for lower fraction of original data
- Augmentations outperform original model on validation data



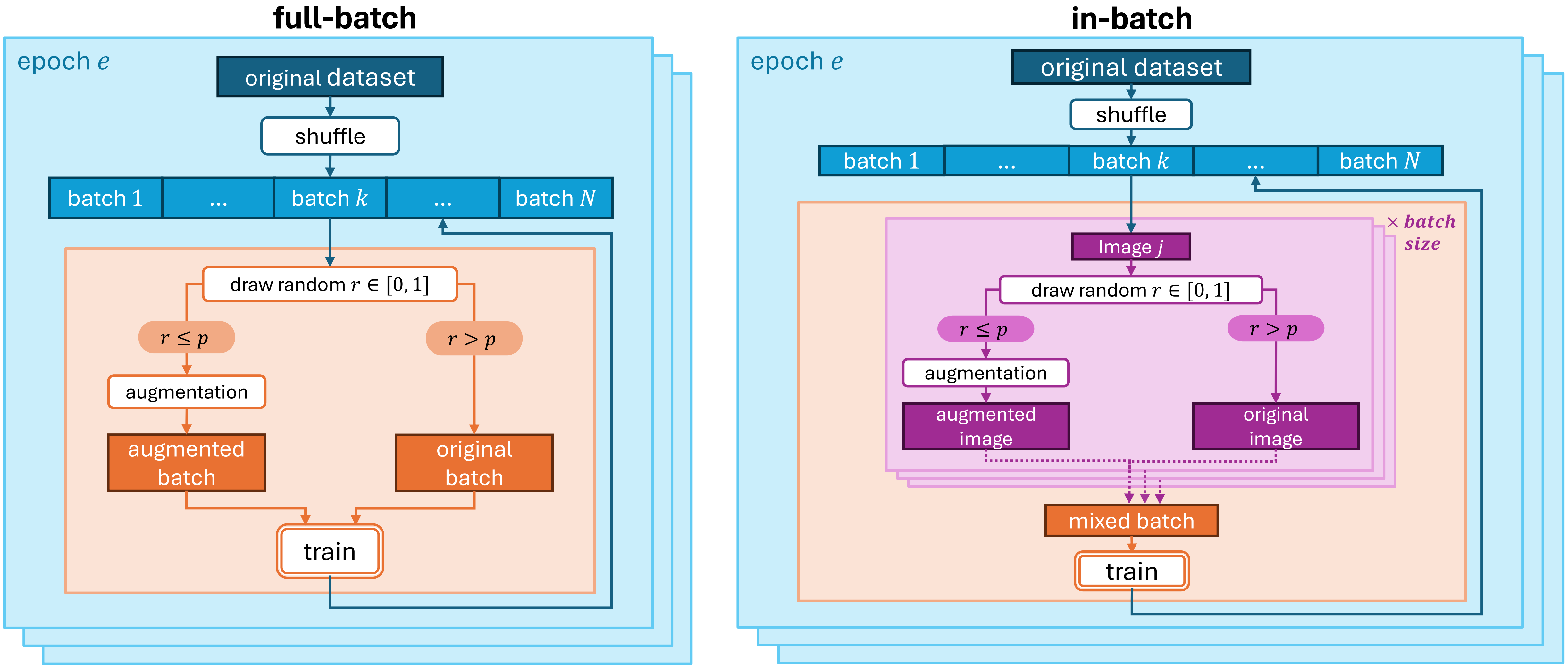


Model	Original	50% aug.	66% aug.	80% aug.	90% aug.	Excl. aug.
Final Acc. [%]	99.3	93.5	91.8	89.7	90.7	90.3

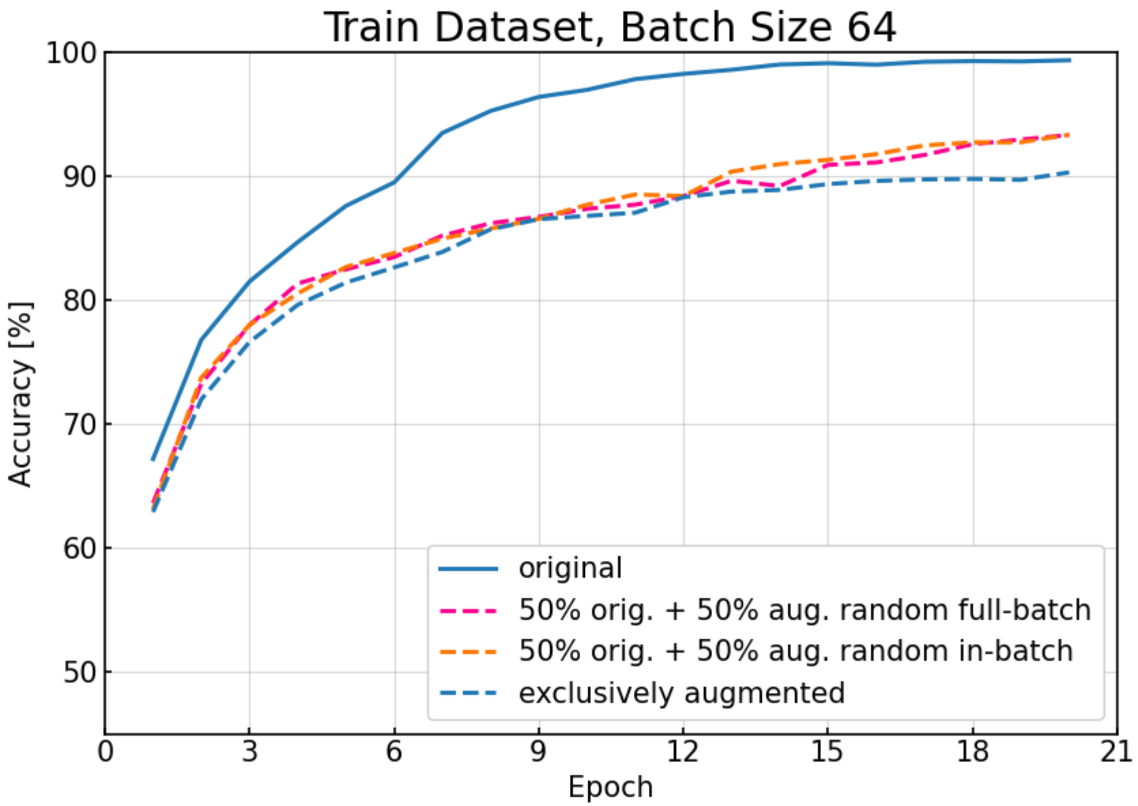
Model	Original	50% aug.	66% aug.	80% aug.	90% aug.	Excl. aug.
Final Acc. [%]	86.5	90.5	89.3	88.8	90.4	88.1

Data mixing: Random

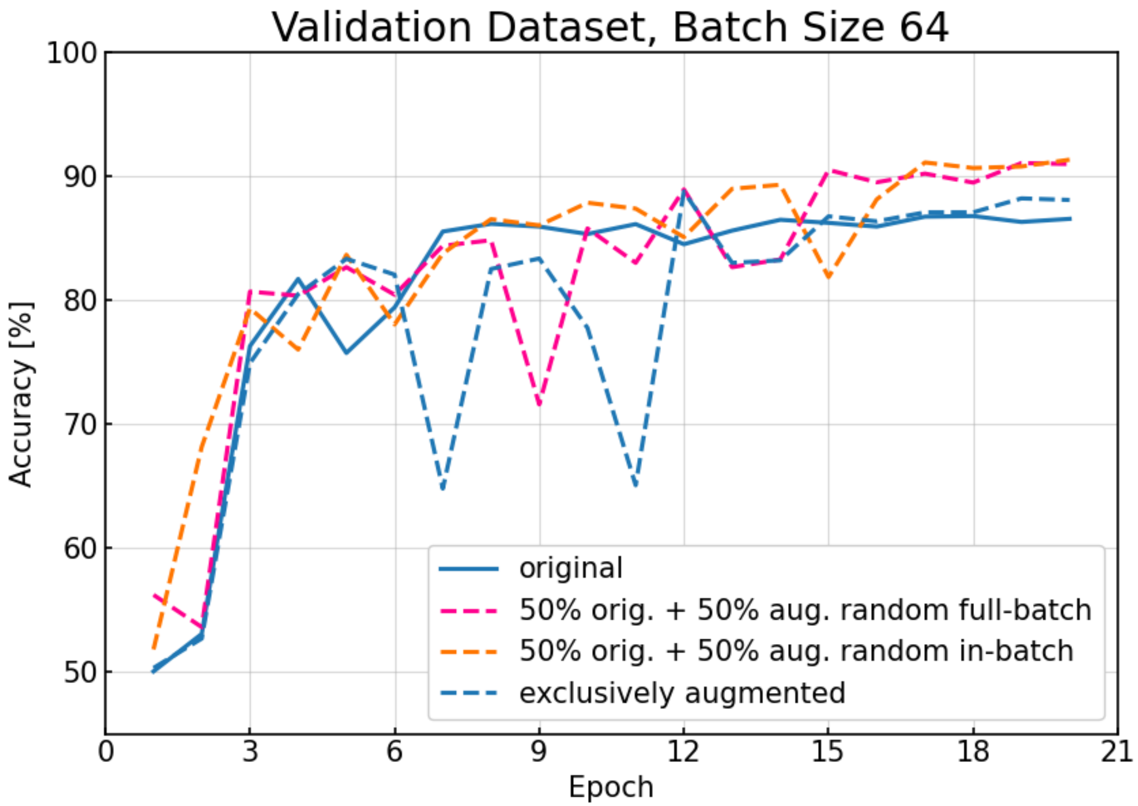
- Free parameter p: fraction of dataset to augment in each epoch



Data mixing: Random



Model	Original	50% aug. full-batch	50% aug. in-batch	Exclusively augmented
Final Acc. [%]	99.3	93.3	93.3	90.3



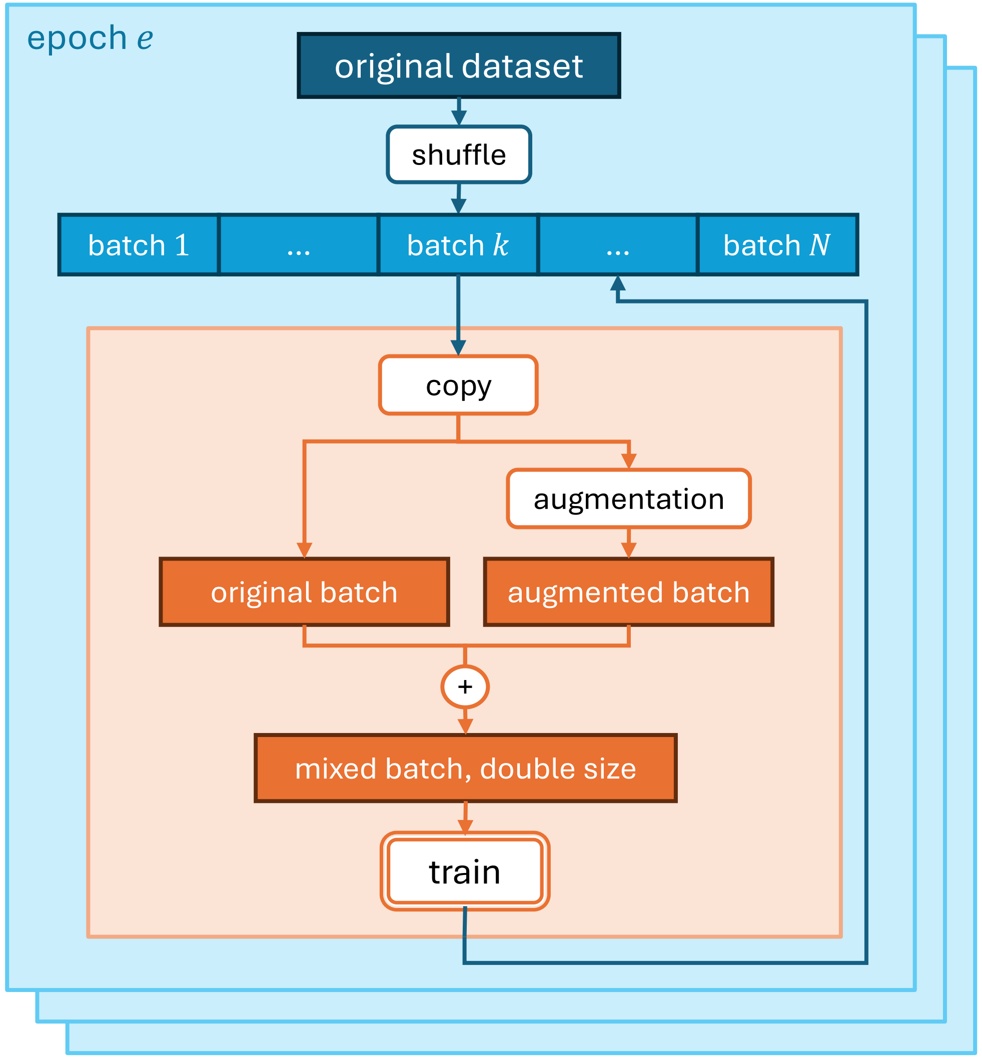
Model	Original	50% aug. full-batch	50% aug. in-batch	Exclusively augmented
Final Acc. [%]	86.5	91.0	91.3	88.1

Augmentation variations

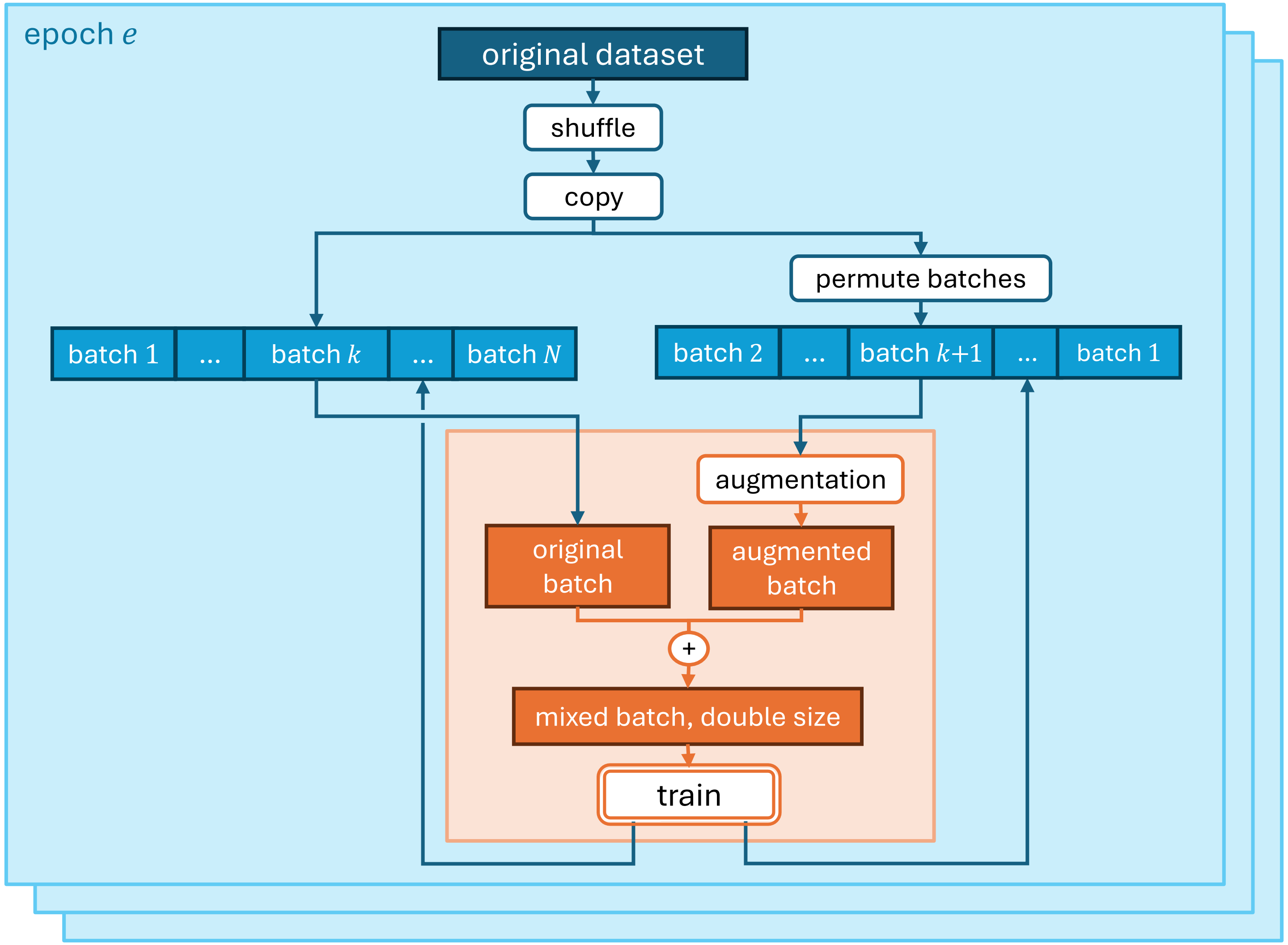
Exclusively augmented	Data mixing	Data doubling
Idea: • Apply augmentation to the full dataset	Idea: • Apply augmentation to a fraction of the dataset Use case: • Modify distribution of training data	Idea: • Use full original dataset • Combine with augmented copy of dataset Use case: • Increase volume of training data

Data doubling

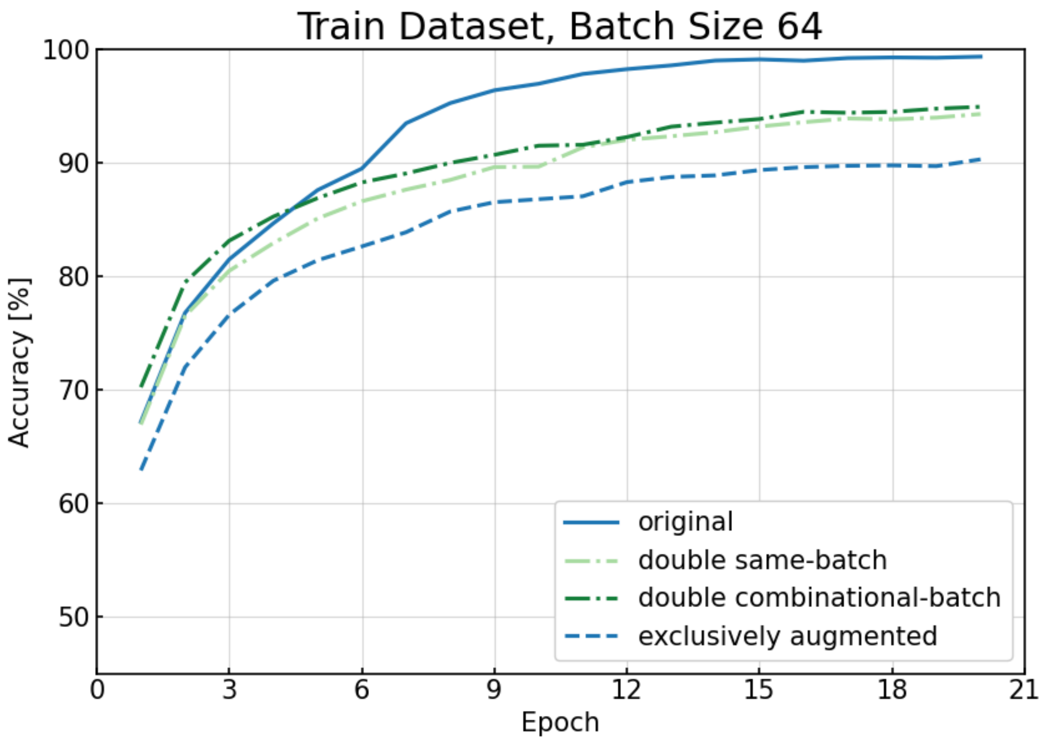
same-batch



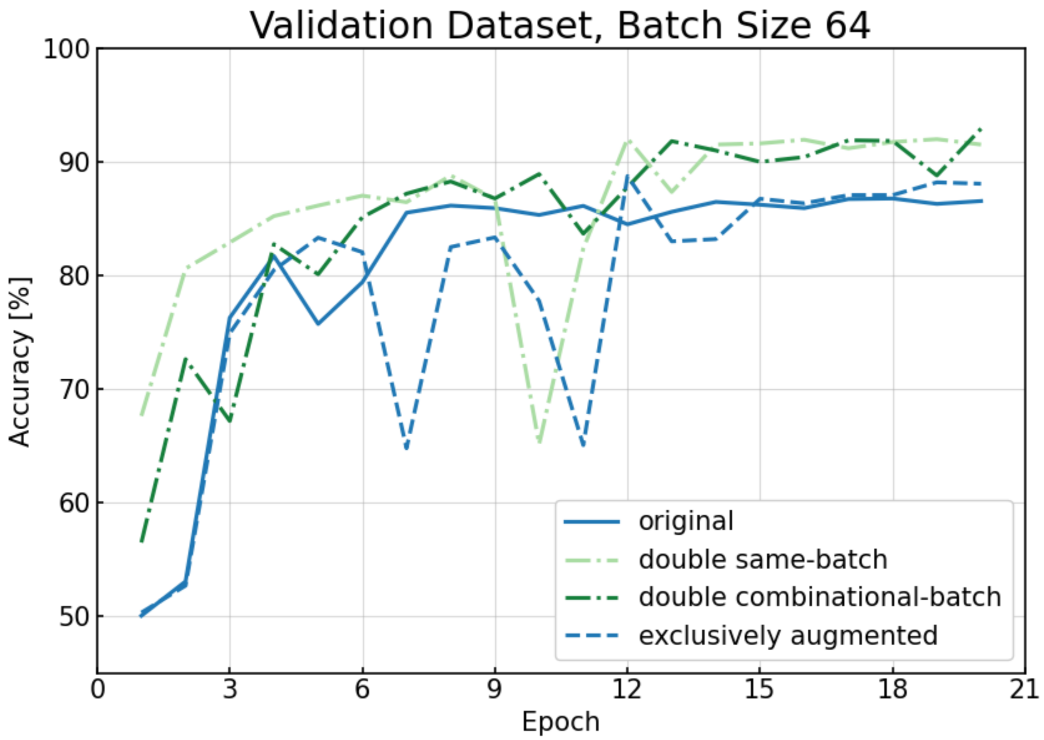
combinational-batch



Data doubling

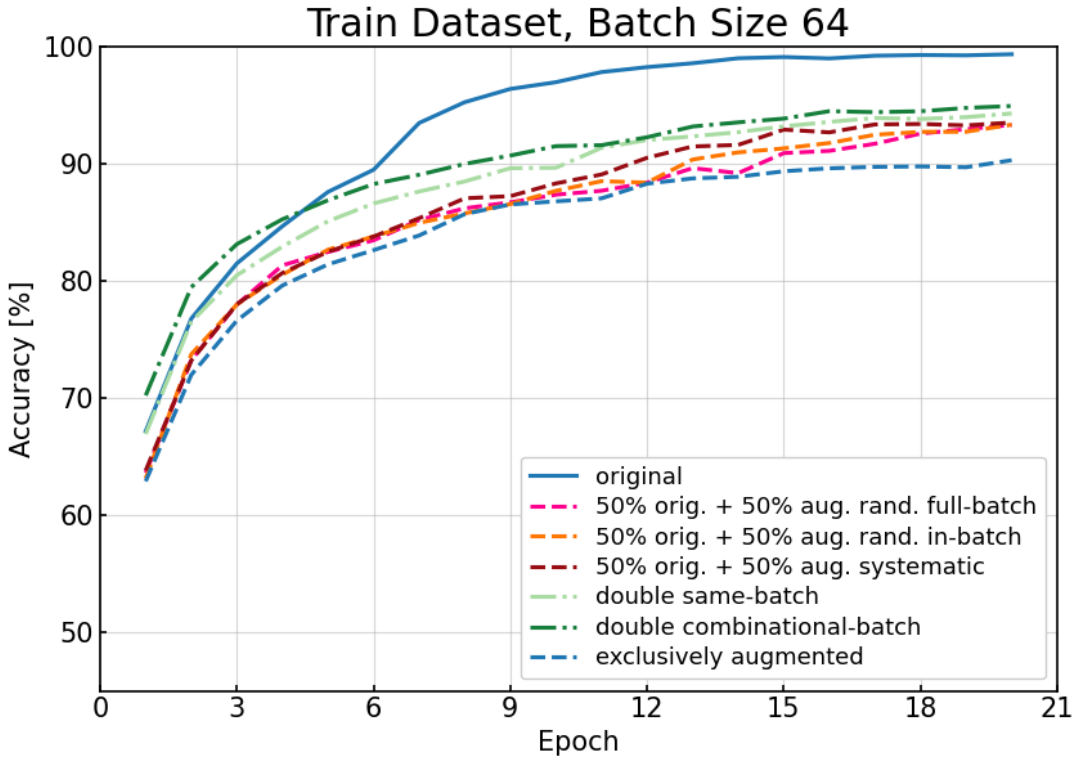


Model	Original	Double same-batch	Double comb.-batch	Exclusively augmented
Final Acc. [%]	99.3	94.3	94.9	90.3

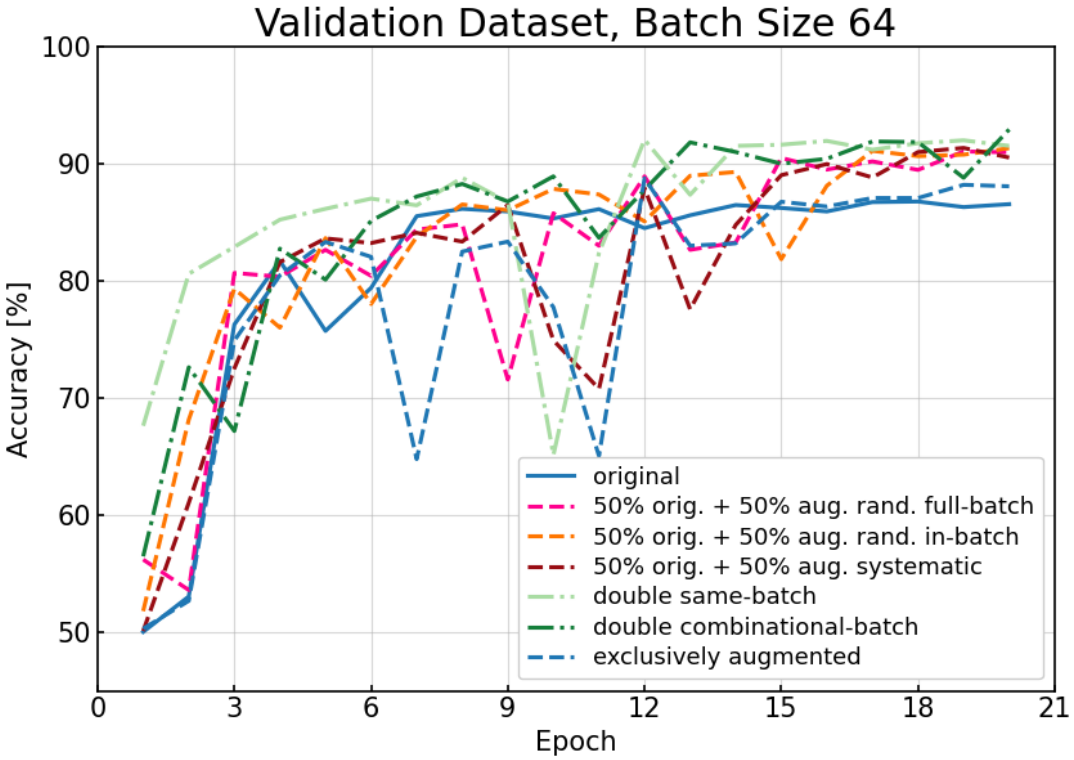


Model	Original	Double same-batch	Double comb.-batch	Exclusively augmented
Final Acc. [%]	86.5	91.5	92.9	88.1

Data augmentation – Summary



Model	Original	50% aug. random full-batch	50% aug. random in-batch	50% aug. syst.	Double same-batch	Double comb.-batch	Excl. aug
Final Acc. [%]	99.3	93.3	93.3	93.5	94.3	94.9	90.3



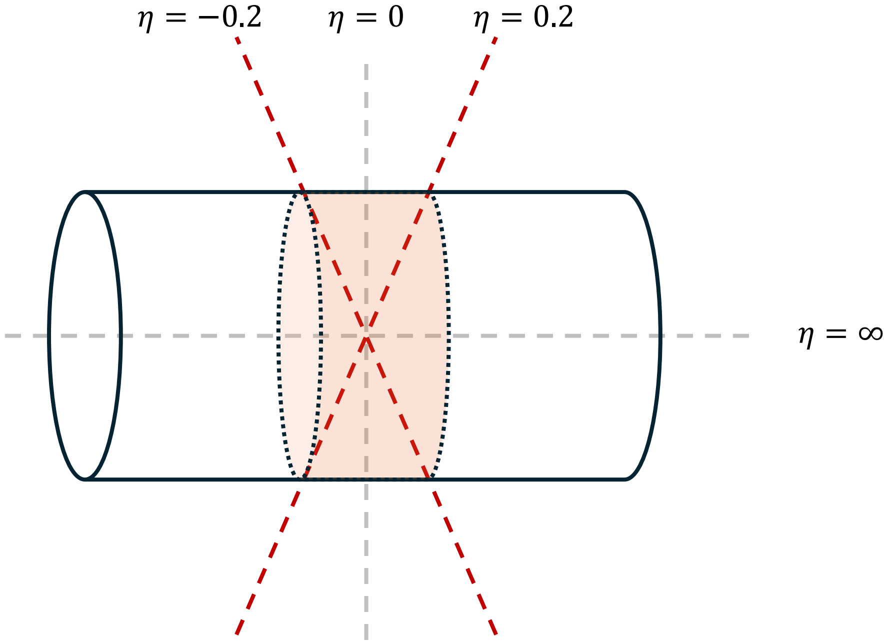
Model	Original	50% aug. random full-batch	50% aug. random in-batch	50% aug. syst.	Double same-batch	Double comb.-batch	Excl. aug
Final Acc. [%]	86.5	91.0	91.3	90.5	91.5	92.9	88.1

Conclusion

- Data augmentation increases validation accuracy
  - Higher statistics of training data
- Larger accuracy improvements for doubling
  - Faster convergence
- Best results from doubling with combinational-batch approach

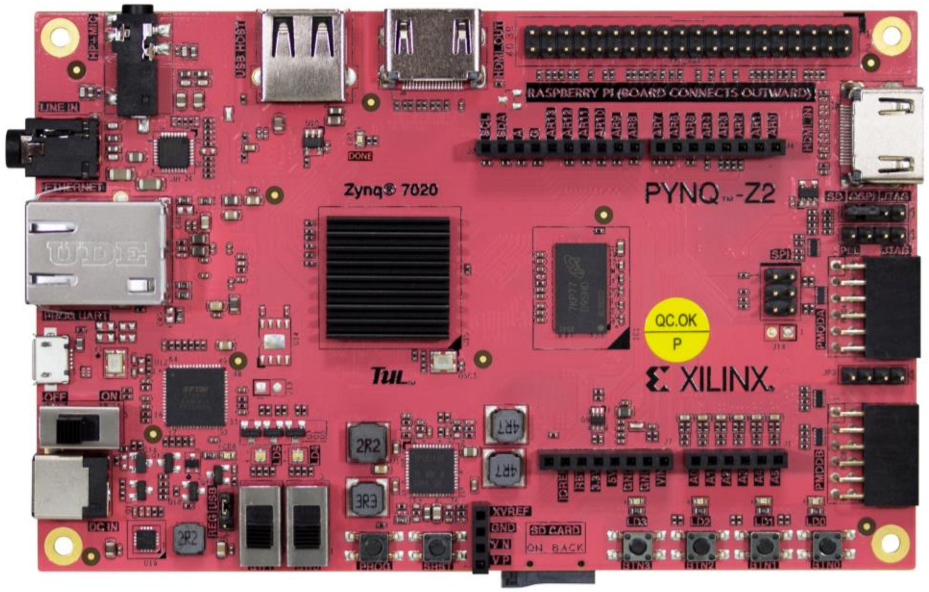
Next steps

- Model training on CMS data
  - Module selection
    - First barrel layer
    - $\eta \in [-0.2, 0.2]$
  - Data: pixel hit map
    - Raw detector data
  - Label: *particle* or *background*
    - From event reconstruction
    - Check for tracks in selected modules



Next steps

- Inference on FPGA board PYNQ-Z2
  - Model size reduction
    - Architecture
    - Pruning
    - Quantization



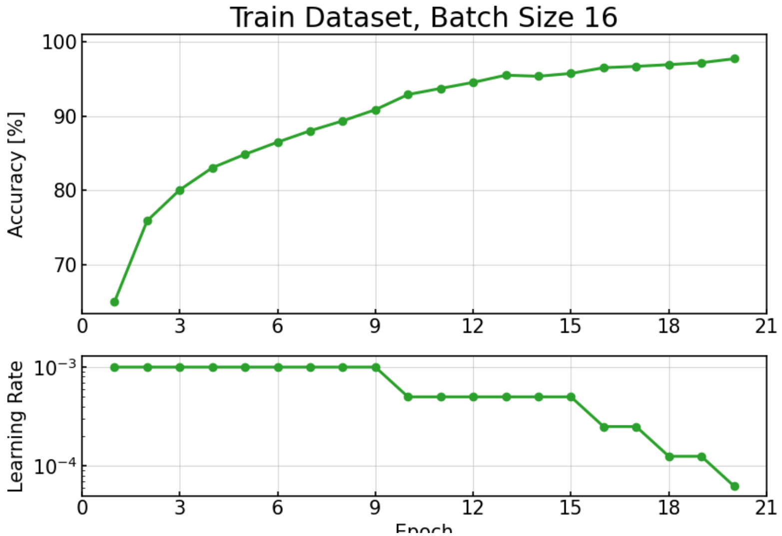
Reduced Model
HLS Resource Estimation

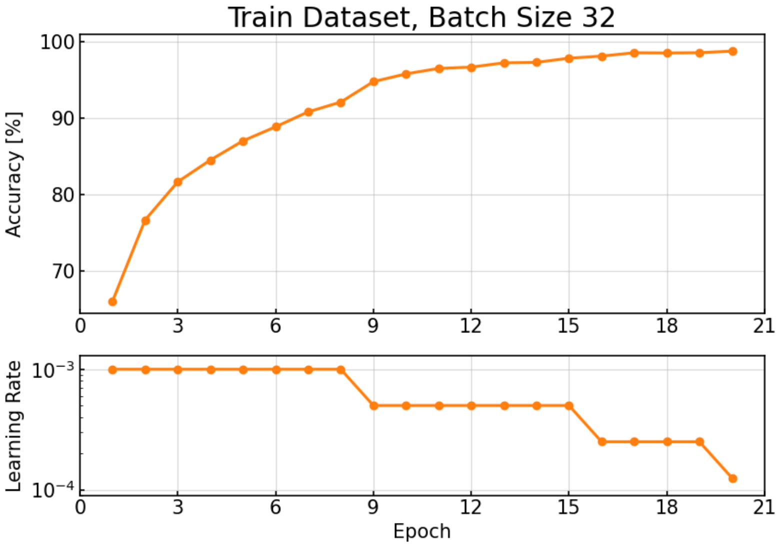
Input	• Shape: (128, 128, 3)
Convolution 2D	• Filters: 16 • Kernel size: 3x3 • Activation: ReLU
Batch Normalisation	
Max Pooling 2D	• Size: 8x8
Dropout	• Rate: 0.25
Flatten	
Dense	• Neurons: 32 • Activation: ReLU
Dropout	• Rate: 0.5
Dense	• Neurons: 1 • Activation: sigmoid

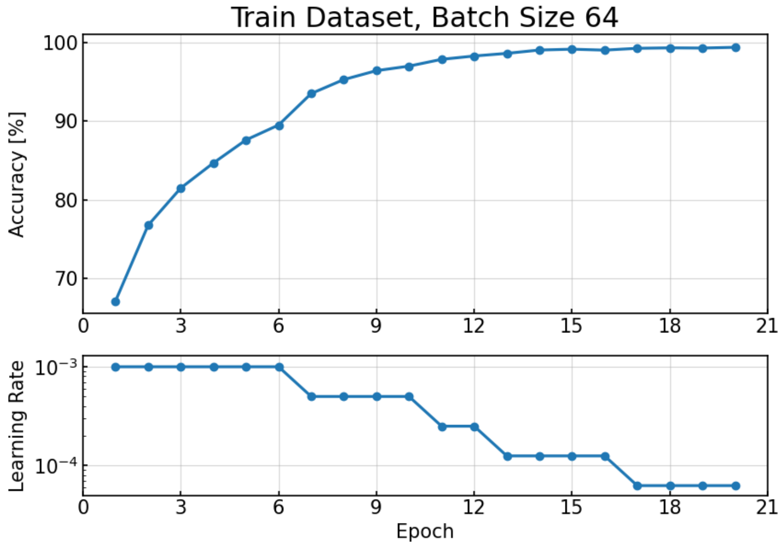
	BRAM	DSP	FF	LUT
Estimated Usage	1171	48	199891	200085
Available	280	220	106400	53200
Utilization (%)	**418**	**21**	**187**	**376**

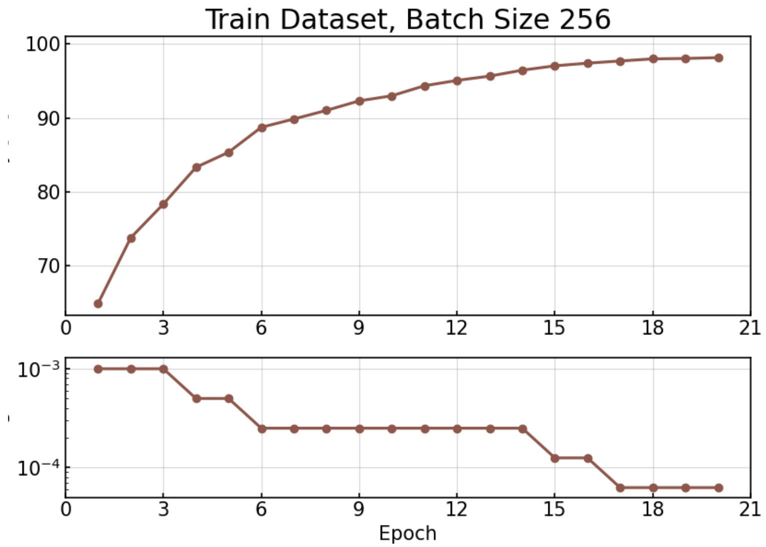
Backup Slides

Base Model Learning Rate









Number of model parameters

Base model

Layer	Output shape	Number of parameters
conv2d_1	(None, 126, 126, 32)	896
batch_normalization_1	(None, 126, 126, 32)	128
max_pooling2d_1	(None, 63, 63, 32)	0
dropout_1	(None, 63, 63, 32)	0
conv2d_2	(None, 61, 61, 64)	18496
batch_normalization_2	(None, 61, 61, 64)	256
max_pooling2d_2	(None, 30, 30, 64)	0
dropout_2	(None, 30, 30, 64)	0
conv2d_3	(None, 28, 28, 128)	73856
batch_normalization_3	(None, 28, 28, 128)	512
max_pooling2d_3	(None, 14, 14, 128)	0
dropout_3	(None, 14, 14, 128)	0
flatten	(None, 25088)	0
dense_1	(None, 512)	12845568
batch_normalization_4	(None, 512)	2048
dropout_4	(None, 512)	0
dense_2	(None, 1)	513
		12942273

Reduced model

Layer	Output shape	Number of parameters
conv2d_1	(None, 126, 126, 16)	448
batch_normalization_1	(None, 126, 126, 16)	64
max_pooling2d_1	(None, 15, 15, 16)	0
dropout_1	(None, 15, 15, 16)	0
flatten	(None, 3600)	0
dense_1	(None, 32)	115232
batch_normalization_2	(None, 32)	128
dropout_2	(None, 32)	0
dense_2	(None, 1)	33
		115905