

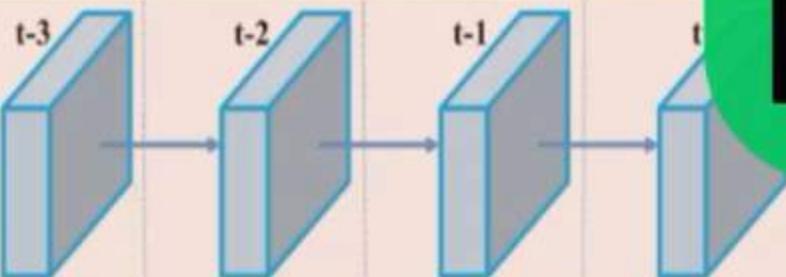
BN [32] – assume a batch contains four mini-batches

accumulate W<sup>(0,3)</sup>  
calculate BN<sup>(0,3)</sup>  
normalize BN

accumulate W<sup>(0,3 ~ 1,2)</sup>  
calculate BN<sup>(0,2)</sup>  
normalize BN

accumulate W<sup>(0,3 ~ 1,2)</sup>  
calculate BN<sup>(0,1)</sup>  
normalize BN

accumulate W<sup>(0,3 ~ 1,2)</sup>  
calculate BN<sup>(0,1)</sup>  
normalize BN  
update W,



# Part-3

CmBN – assume a batch contains four mini-batches

accumulate W<sup>(0,3)</sup>  
accumulate W<sup>(0,3 ~ 1,2)</sup>  
normalize BN

# YOLO-V4

P<sub>7</sub> C

P<sub>6</sub> O

P<sub>5</sub> O

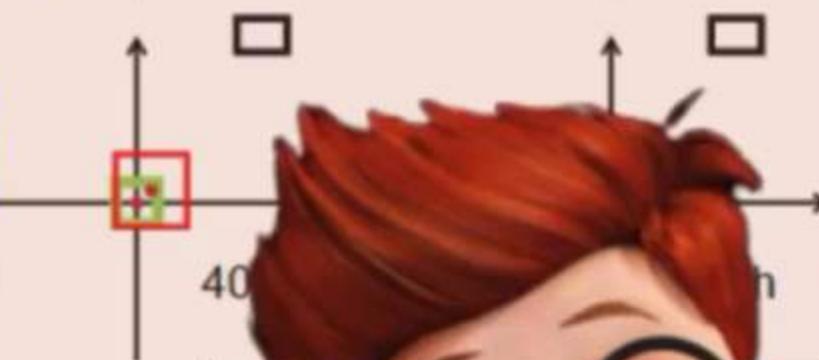
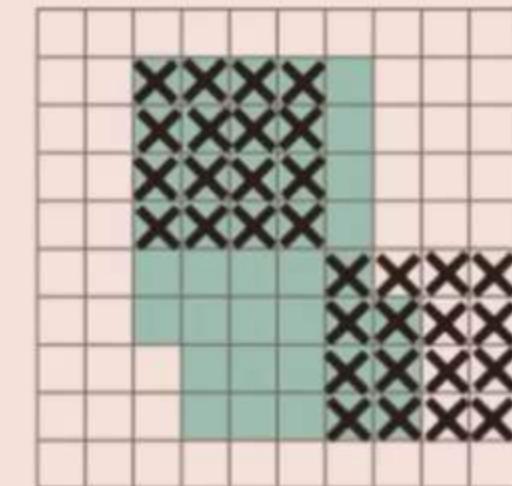
P<sub>4</sub> O

P<sub>3</sub> O

P<sub>5</sub> O



- CmBN, MiWRC
- Drop Block, CIoU, DIoU



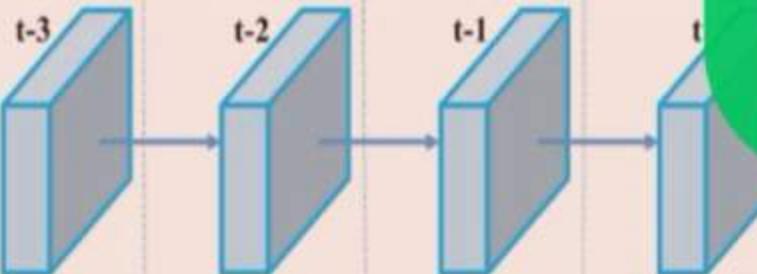
BN [32] – assume a batch contains four mini-batches

accumulate W<sup>(0,3)</sup>  
calculate BN<sup>(0,3)</sup>  
normalize BN

accumulate W<sup>(0,3 ~ 1,2)</sup>  
calculate BN<sup>(0,2)</sup>  
normalize BN

accumulate W<sup>(0,3 ~ 1,2)</sup>  
calculate BN<sup>(0,1)</sup>  
normalize BN

accumulate W  
calculate BN  
normalize  
update W,



# Part-3

CmBN – assume a batch contains four mini-batches

accumulate W<sup>(0,3)</sup>  
accumulate W<sup>(0,3 ~ 1,2)</sup>  
normalize BN

YOLO-V4

P<sub>7</sub> C

P<sub>6</sub> O

P<sub>5</sub> O

P<sub>4</sub> O

P<sub>3</sub> O

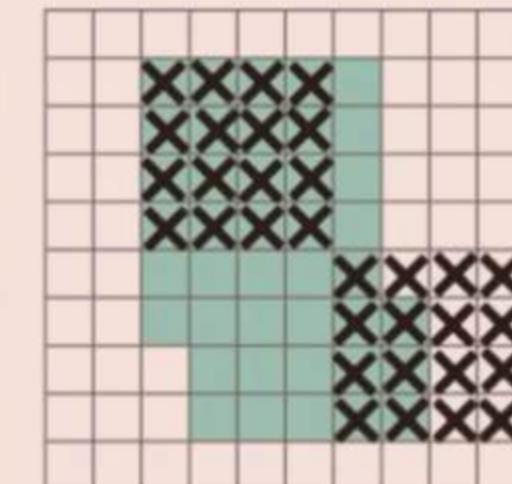
P<sub>5</sub> O



- CmBN, MiWRC
- Drop Block, CIoU, DIoU

(b) PANet

(d) BiFPN



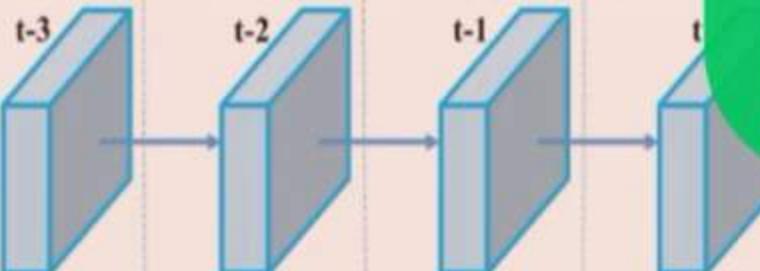
BN [32] – assume a batch contains four mini-batches

accumulate W<sup>(t,3)</sup>  
calculate BN<sup>(t,3)</sup>  
normalize BN

accumulate W<sup>(t,3 ~ t,2)</sup>  
calculate BN<sup>(t,2)</sup>  
normalize BN

accumulate W<sup>(t,3 ~ t,1)</sup>  
calculate BN<sup>(t,1)</sup>  
normalize BN

accumulate W  
calculate BN  
normalize update W.

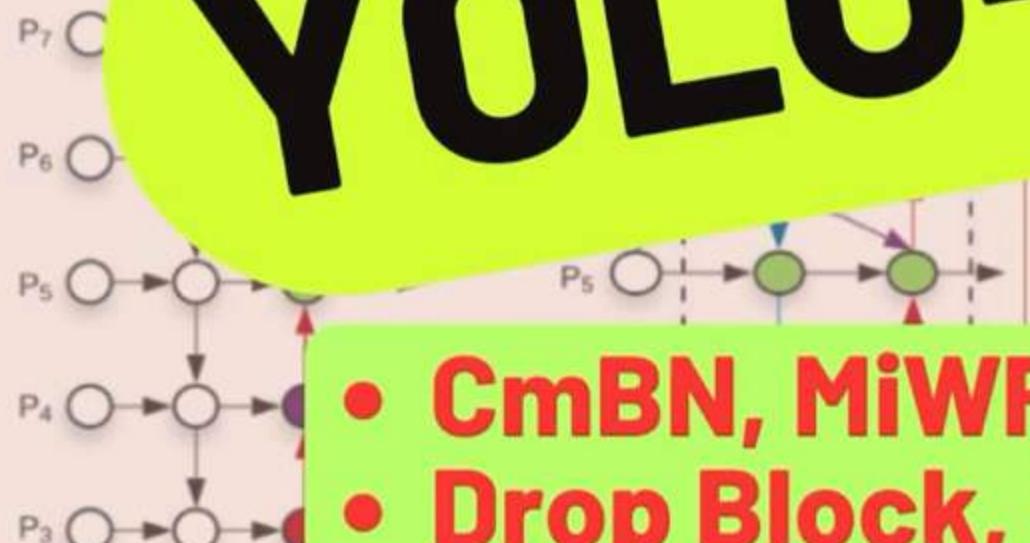


# Part-3

CmBN – assume a batch contains four mini-batches

accumulate W<sup>(t,3)</sup>  
accumulate W<sup>(t,3 ~ t,2)</sup>  
normalize BN

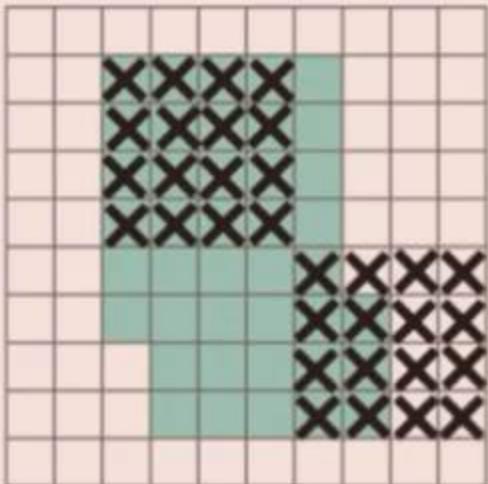
# YOLO-V4



- CmBN, MiWRC
- Drop Block, CIoU, DIoU

(b) PANet

(d) BiFPN



# In Previous video..

- **Backbone**
  - DenseNet
  - CSPNet
  - CSPDarknet-53
- **Neck**
  - FPN
  - SPP
  - PAN
- **Spatial Attention Module**

# In Previous video..

- **Backbone**
  - DenseNet
  - CSPNet
  - CSPDarknet-53
- **Neck**
  - FPN
  - SPP
  - PAN
- **Spatial Attention Module**

# In Previous video..

- **Backbone**
  - DenseNet
  - CSPNet
  - CSPDarknet-53
- **Neck**
  - FPN
  - SPP
  - PAN
- **Spatial Attention Module**

# BoF & BoS - covered

|                              | Backbone  | Detector   |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul>                               | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• Cross-stage partial connections (CSP)</li><li>• Multi-input weighted residual connections (MiWRC)</li></ul> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• SPP-block</li><li>• SAM-block</li><li>• PAN path-aggregation block</li><li>• DIoU-NMS</li></ul>  |

# BoF & BoS - covered

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul>                               | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• Cross-stage partial connections (CSP)</li><li>• Multi-input weighted residual connections (MiWRC)</li></ul> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• SPP-block</li><li>• SAM-block</li><li>• PAN path-aggregation block</li><li>• DIoU-NMS</li></ul>  |

# BoF & BoS - covered

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul>                               | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• Cross-stage partial connections (CSP)</li><li>• Multi-input weighted residual connections (MiWRC)</li></ul> | <ul style="list-style-type: none"><li>• Mish activation</li><li>• SPP-block</li><li>• SAM-block</li><li>• PAN path-aggregation block</li><li>• DIoU-NMS</li></ul>  |

# In this Video

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>   |

# In this Video

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>   |

# In this Video

|                              | Backbone  | Detector  |
|------------------------------|---|---|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini <del>patch</del> Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>  |

# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

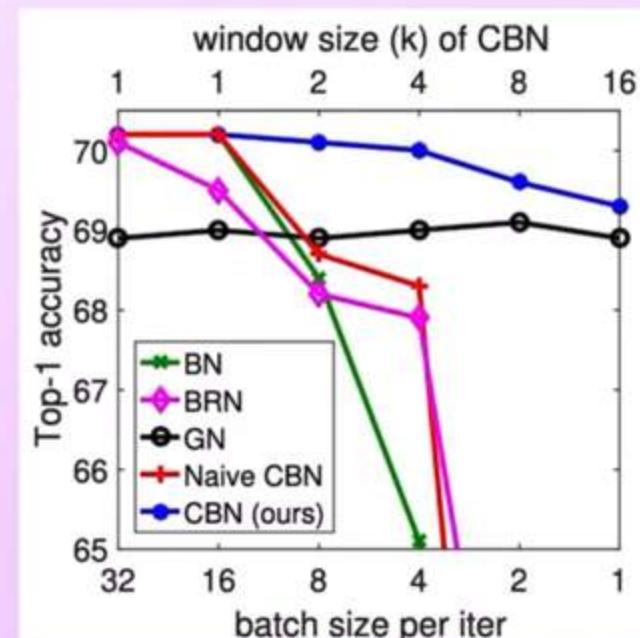
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

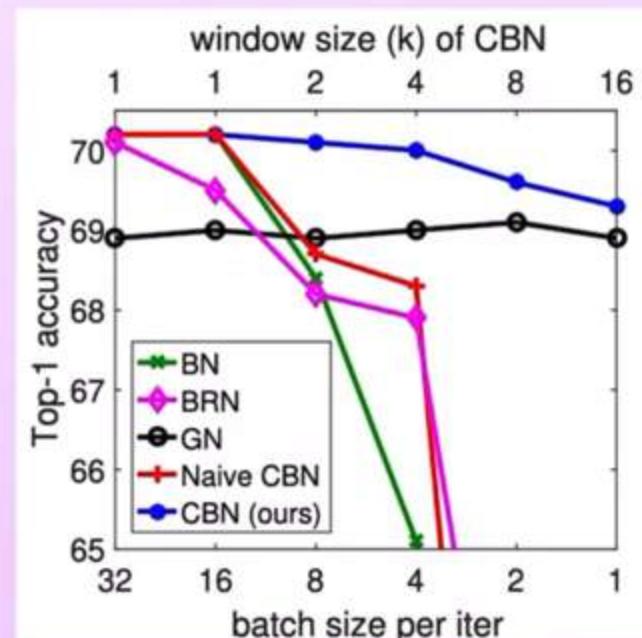
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

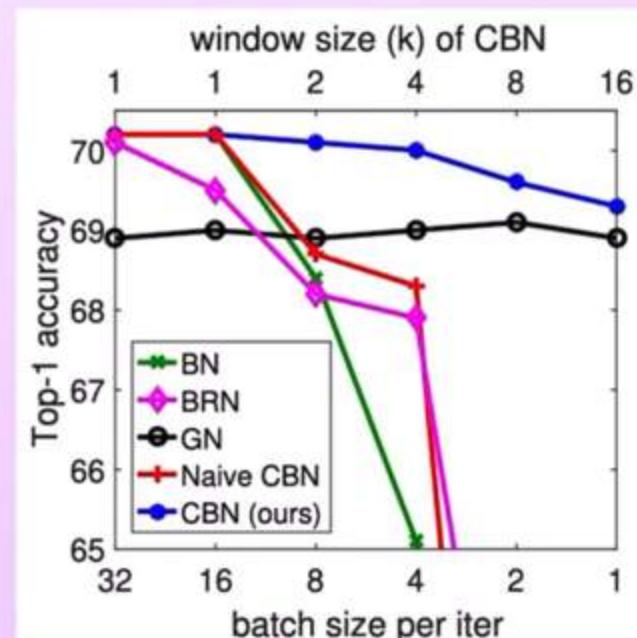
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

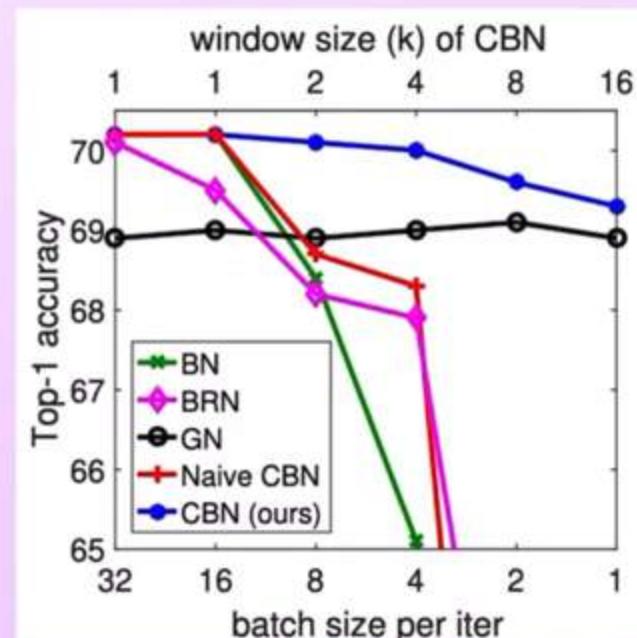
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

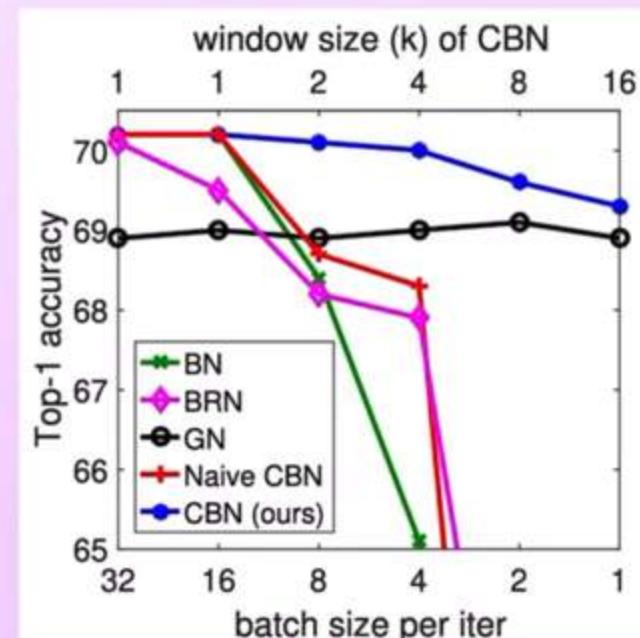
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

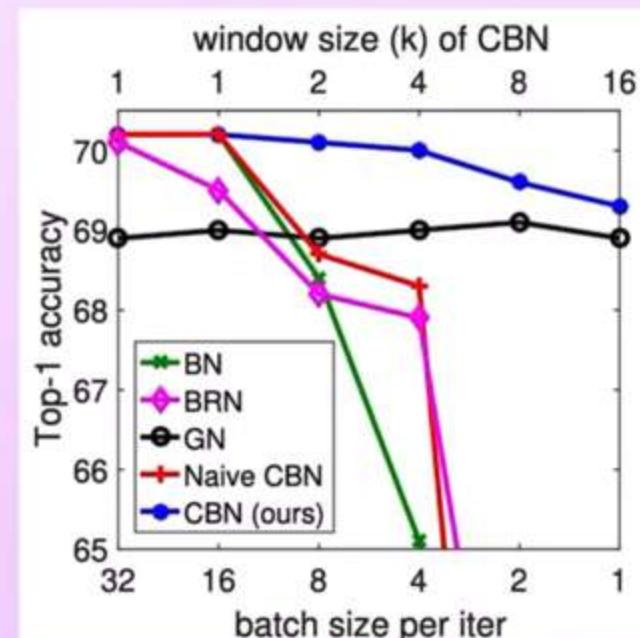
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

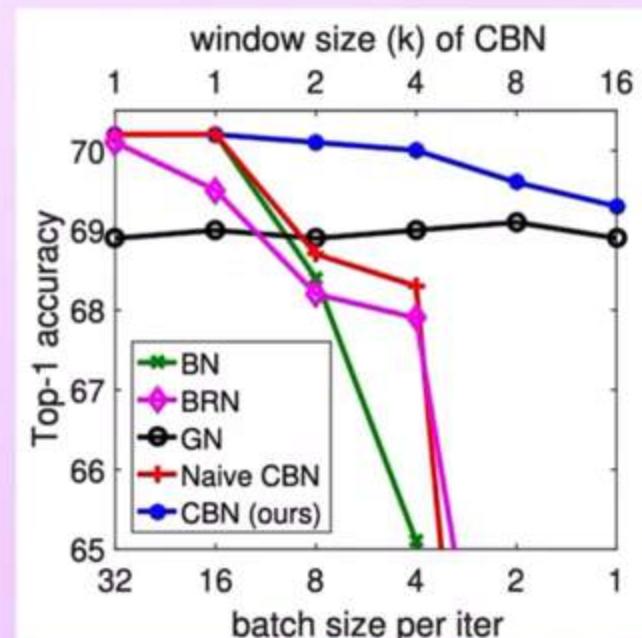
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

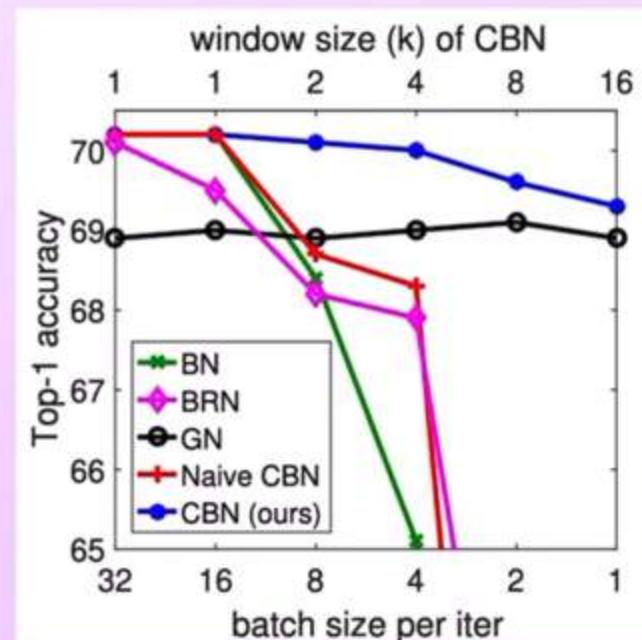
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

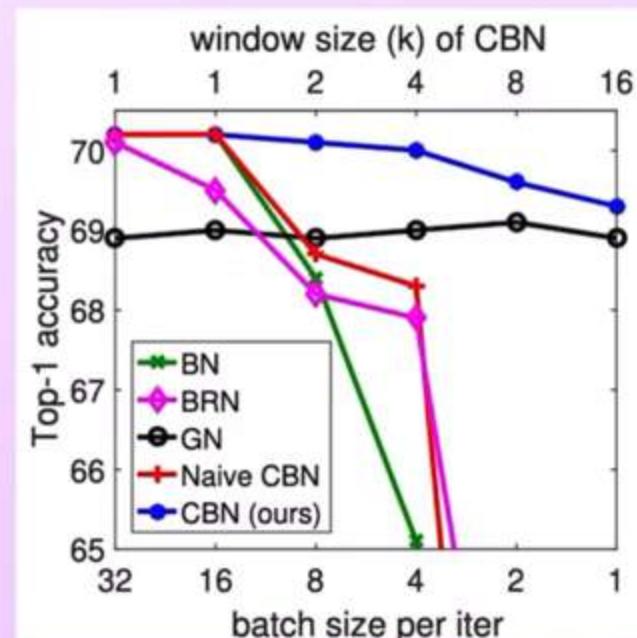
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

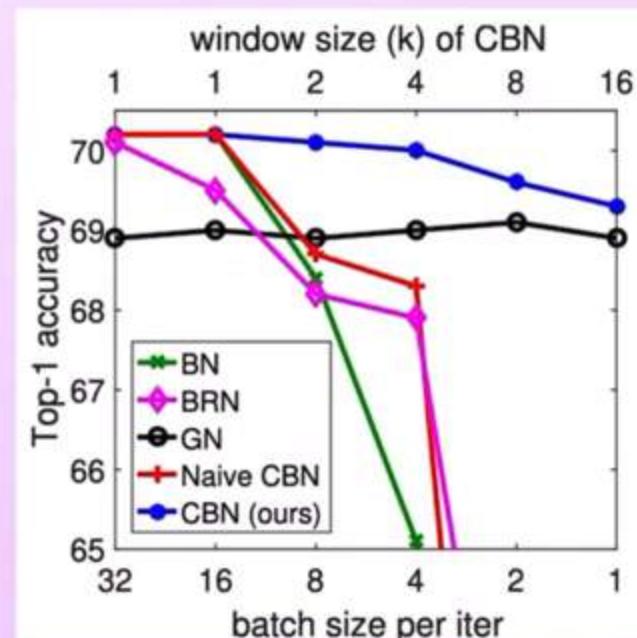
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

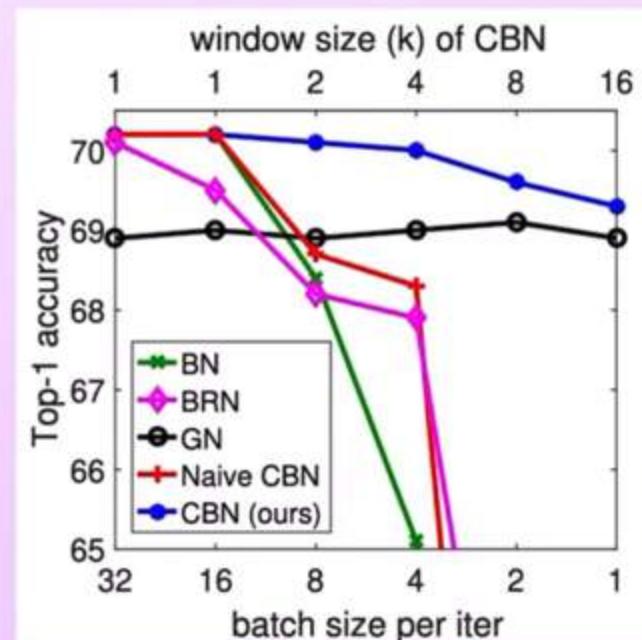
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

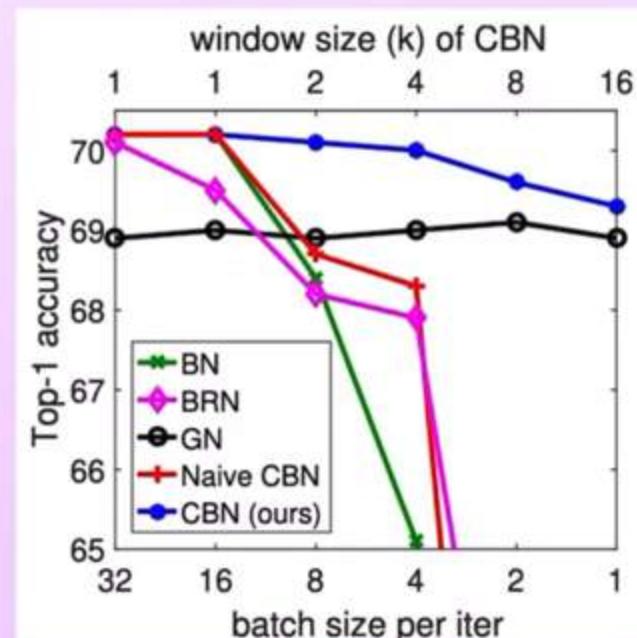
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

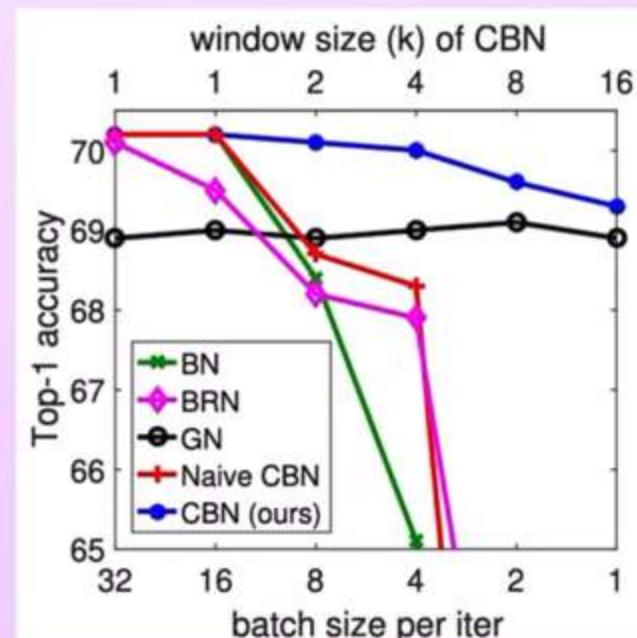
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

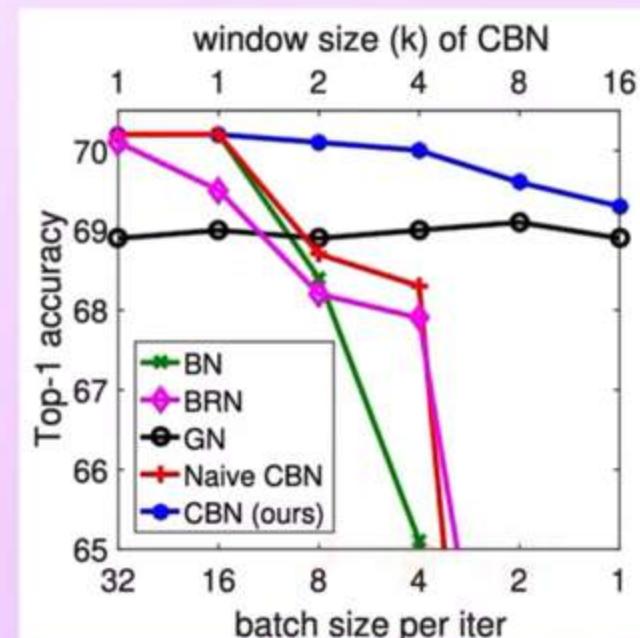
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

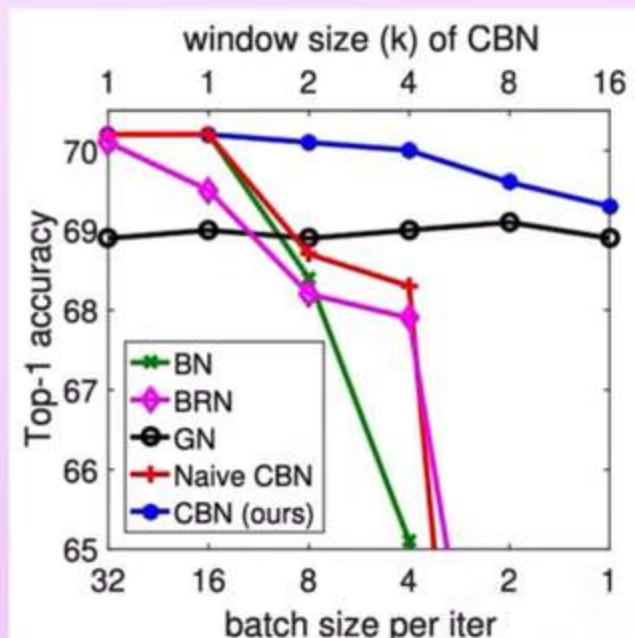
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

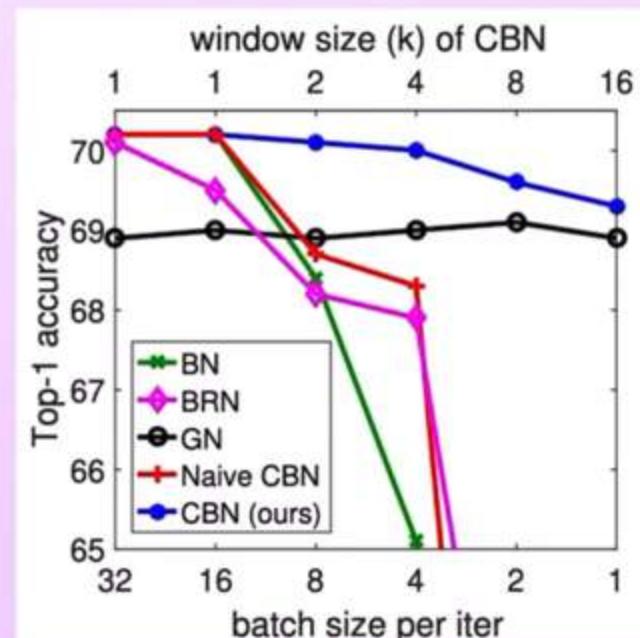
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

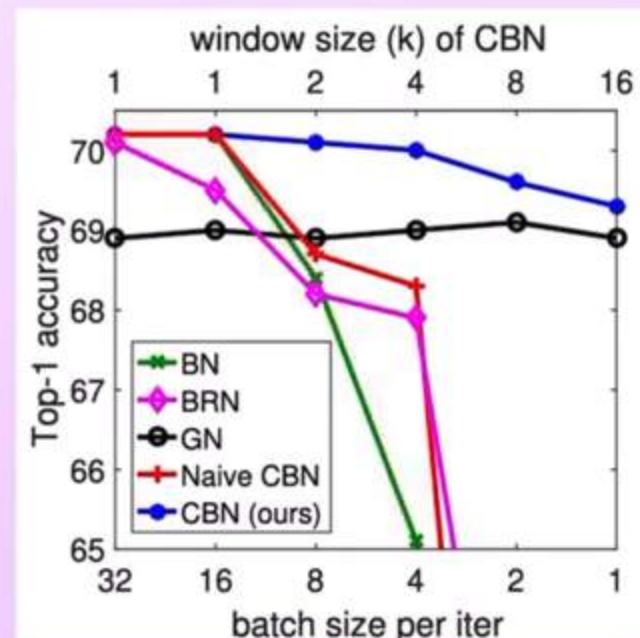
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

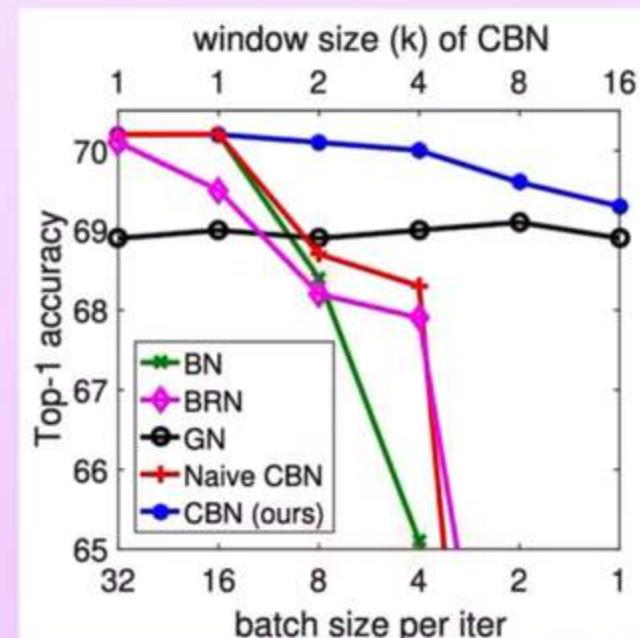
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Batch Normalization:

- Mean ( $\mu$ ):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$$

- Variance ( $\sigma^2$ ):

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_{ij} - \mu_j)^2$$

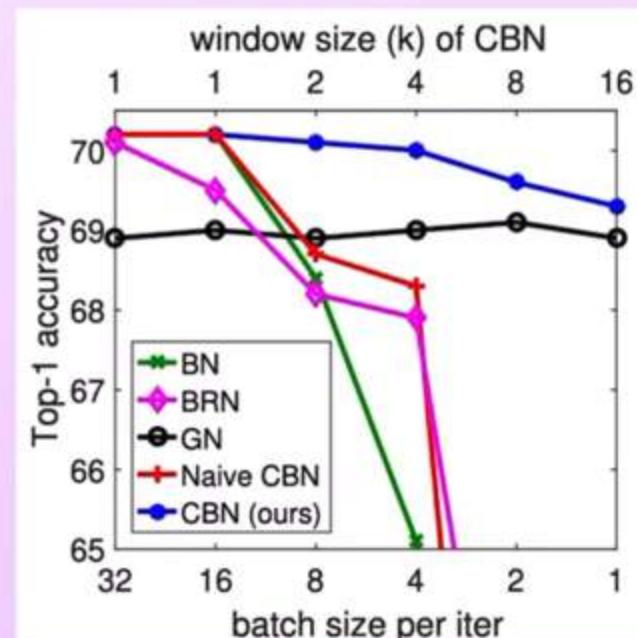
- Normalize:

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

Drawbacks:

- Fails in case of small batch sizes  
(mean and variance are inaccurate)



# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

- . Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

- . Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

. Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

- . Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

- . Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

# Cross Mini Batch Normalization

Idea: What if we take previous batches also for mean and variance calculations

- . Collect statistics over the entire batch instead of mini-batches

- CBN: CmBN - using Cross mini-Batch Normalization for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch

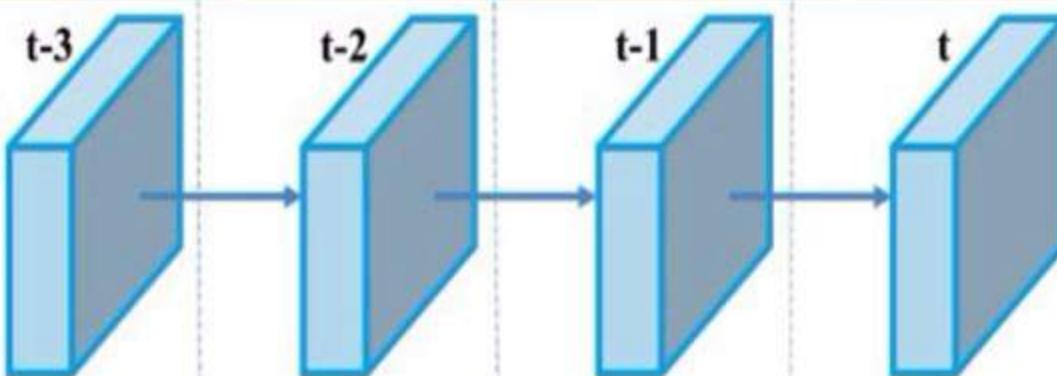
# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |  |
|--|---|---|--|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>calculate $BN^{(t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|--|---|---|--|

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



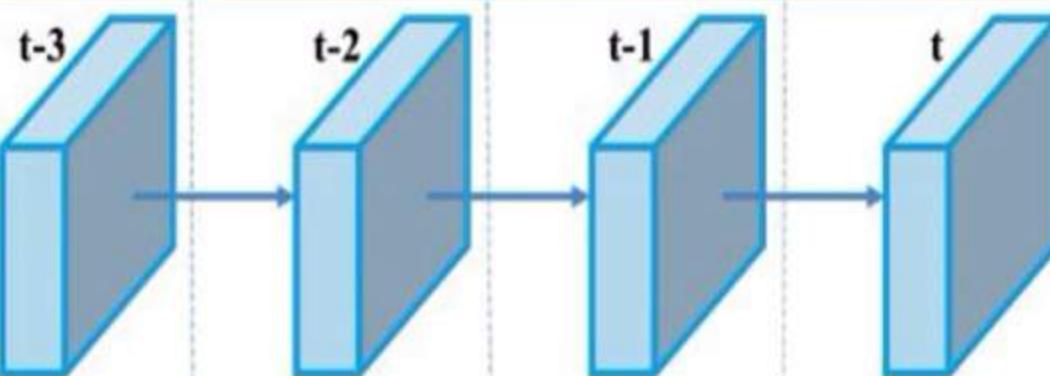
**CmBN – assume a batch contains four mini-batches**

|   |   |   |  |
|---|---|---|--|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|---|---|---|--|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |  |
|--|---|---|--|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>calculate $BN^{(t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|--|---|---|--|



Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W

**CmBN – assume a batch contains four mini-batches**

|   |   |   |  |
|---|---|---|--|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|---|---|---|--|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

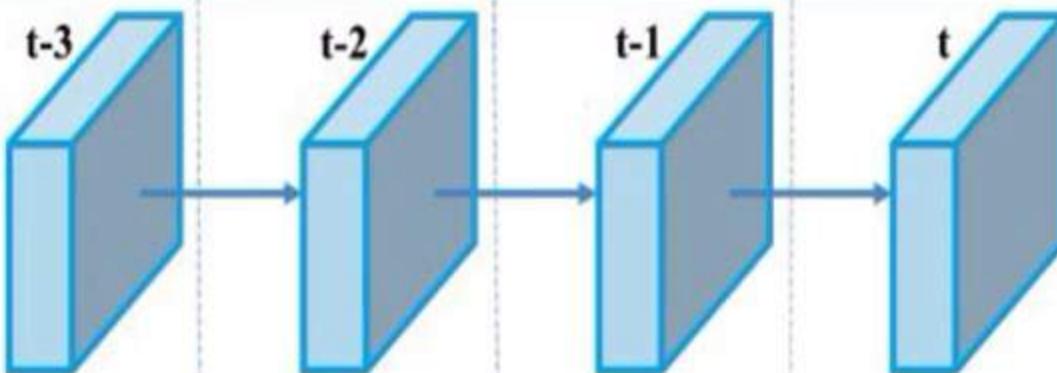
accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

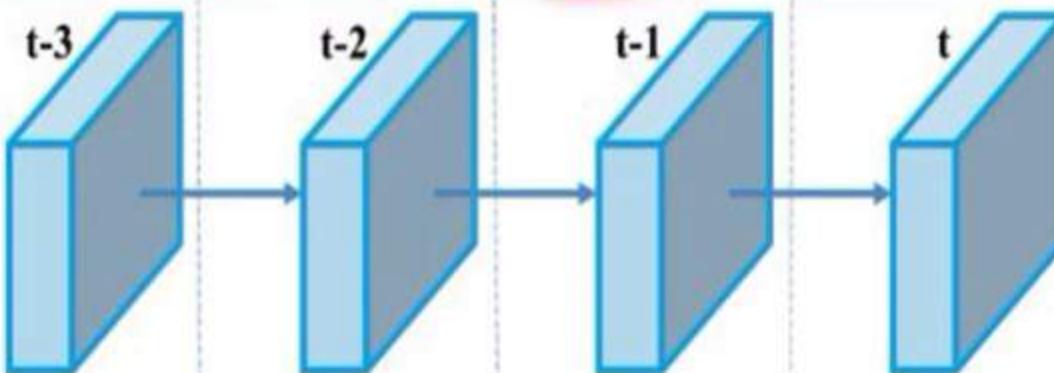
# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |  |
|--|---|---|--|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br><del>calculate <math>BN^{(t-1)}</math></del><br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|--|---|---|--|

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



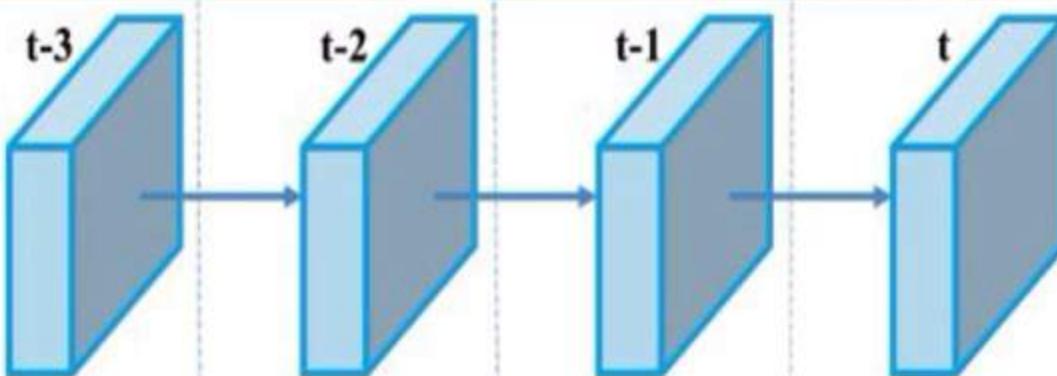
**CmBN – assume a batch contains four mini-batches**

|   |   |   |  |
|---|---|---|--|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|---|---|---|--|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |   |
|--|---|---|---|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>calculate $BN^{(t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update W, ScaleShift |
|--|---|---|---|



**CmBN – assume a batch contains four mini-batches**

|   |   |   |   |
|---|---|---|---|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update W, ScaleShift |
|---|---|---|---|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

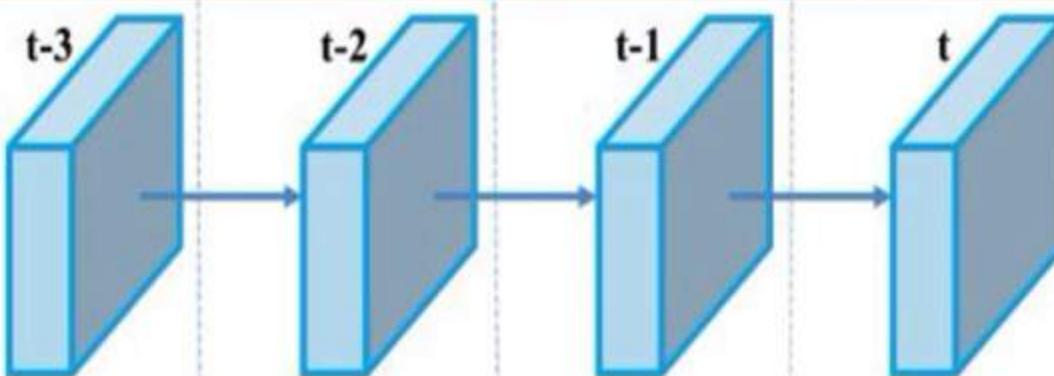
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

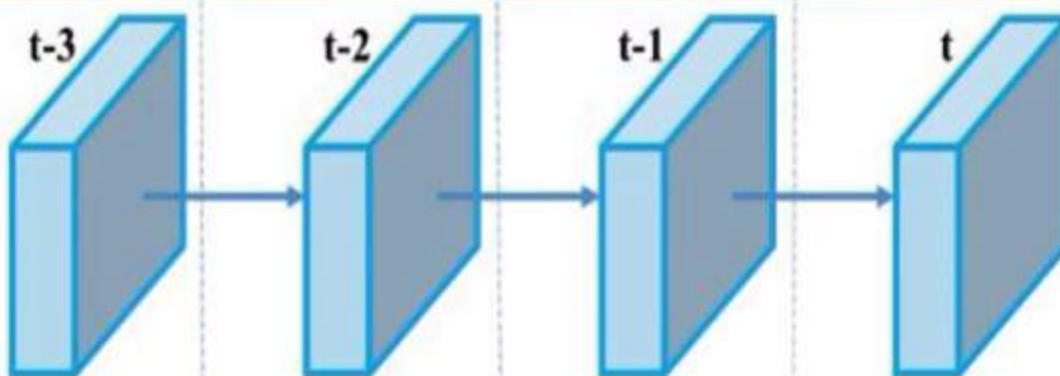
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
~~calculate  $BN^{(t)}$~~   
normalize BN  
update W, ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update W, ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

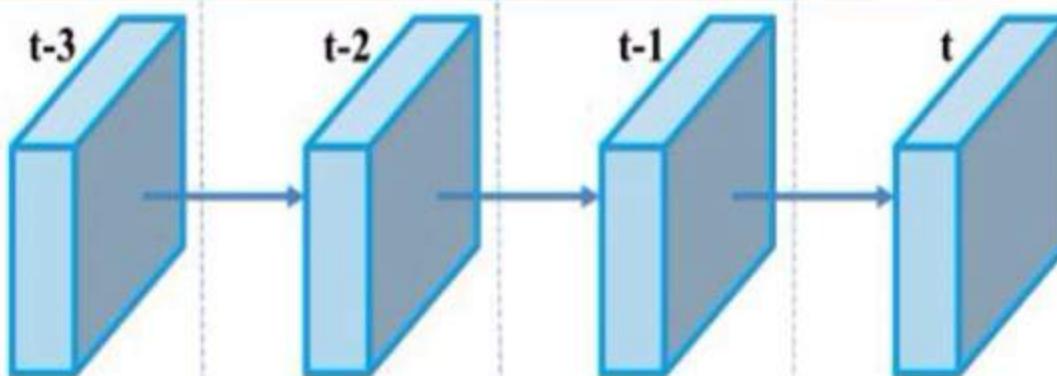
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

BN [32] – assume a batch contains four mini-batches

Q

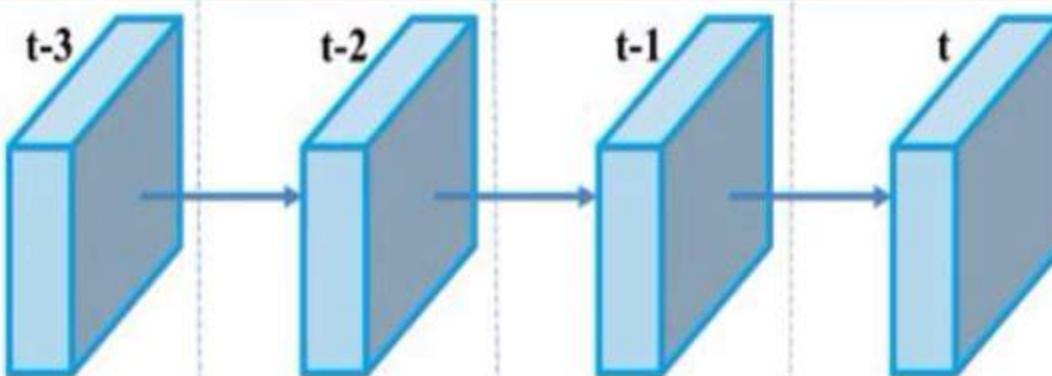
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation



CmBN – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

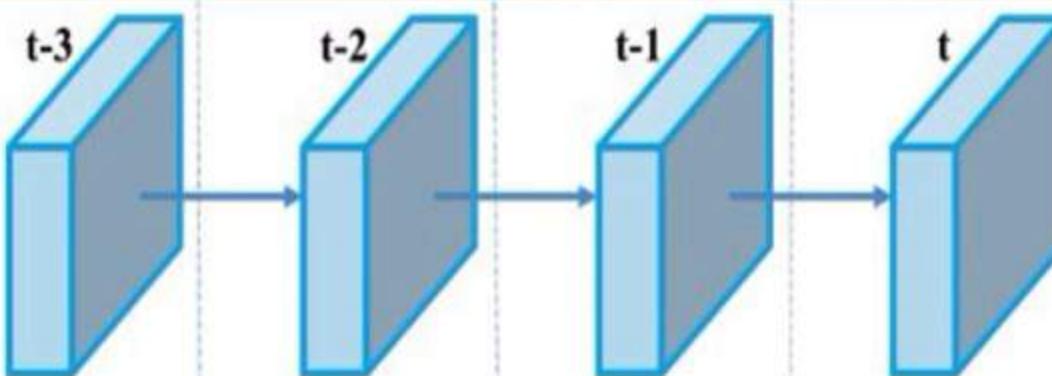
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation



Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W

**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

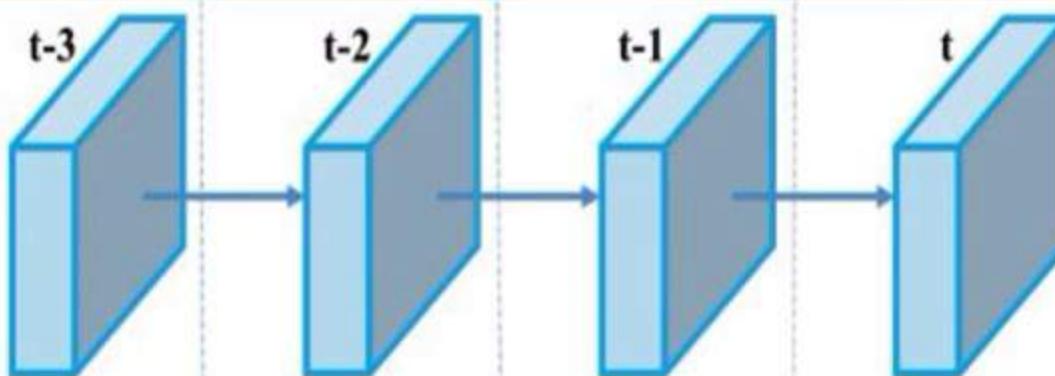
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

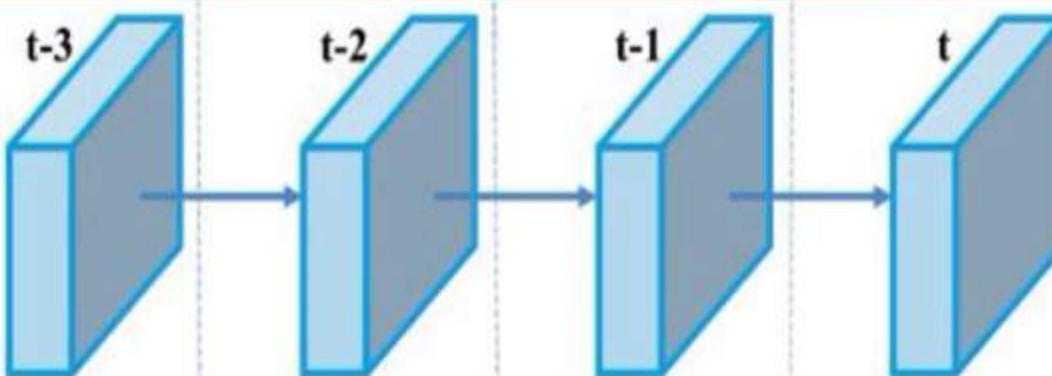
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation



Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W

**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

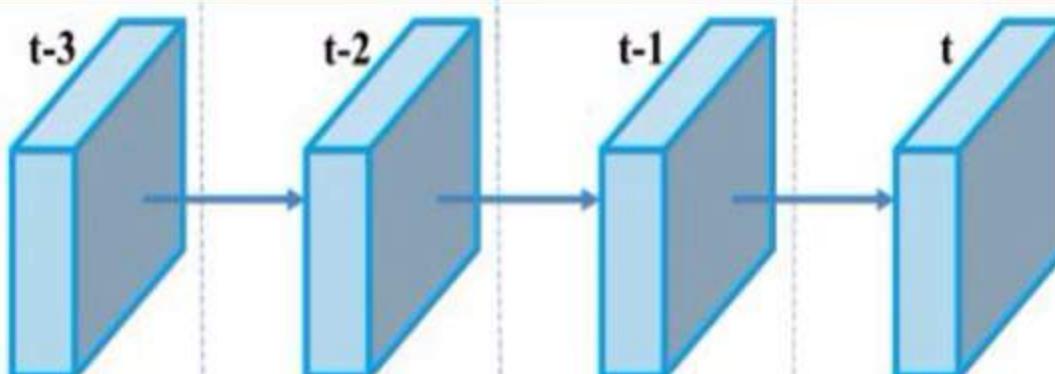
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32]** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

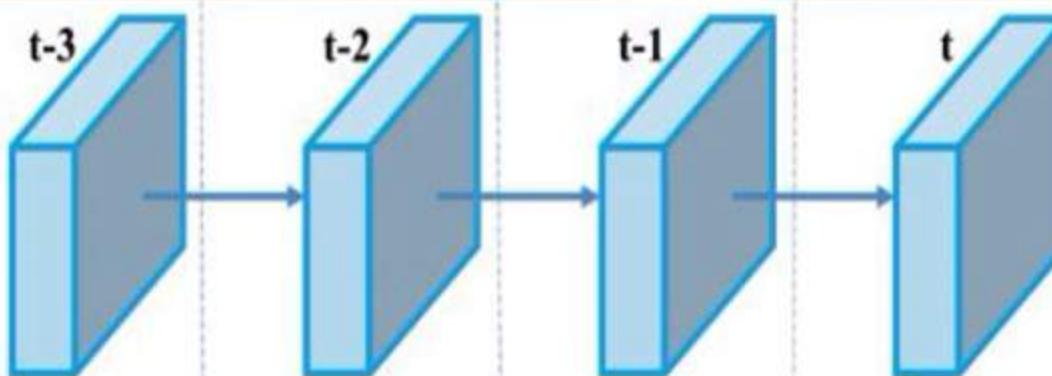
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

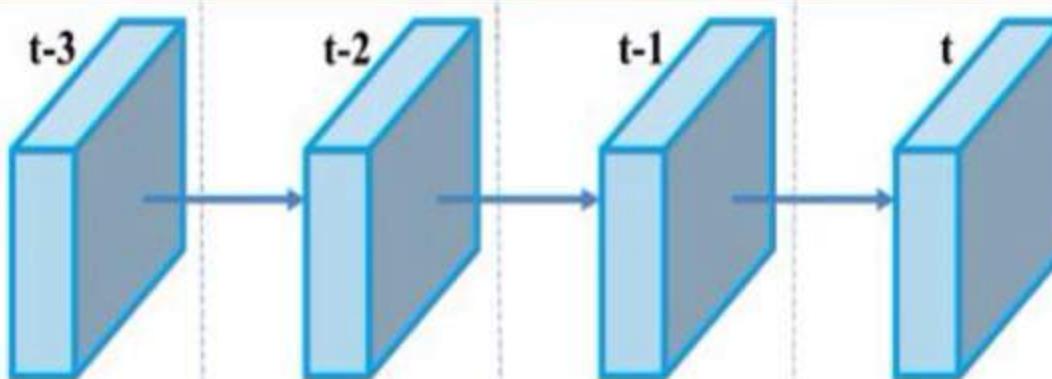
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

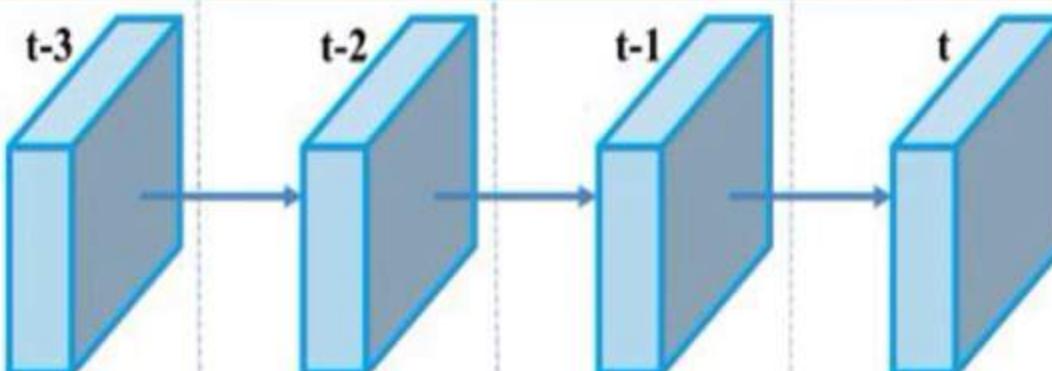
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update W, ScaleShift

Gradient  
Accumulation



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update W, ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

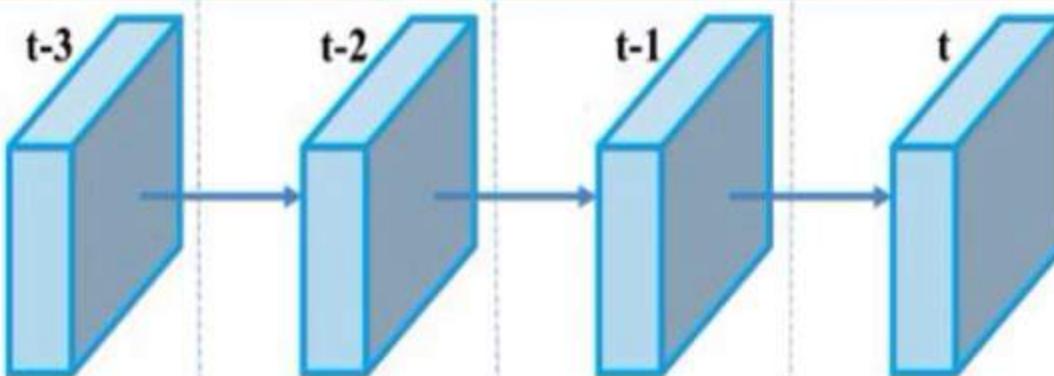
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

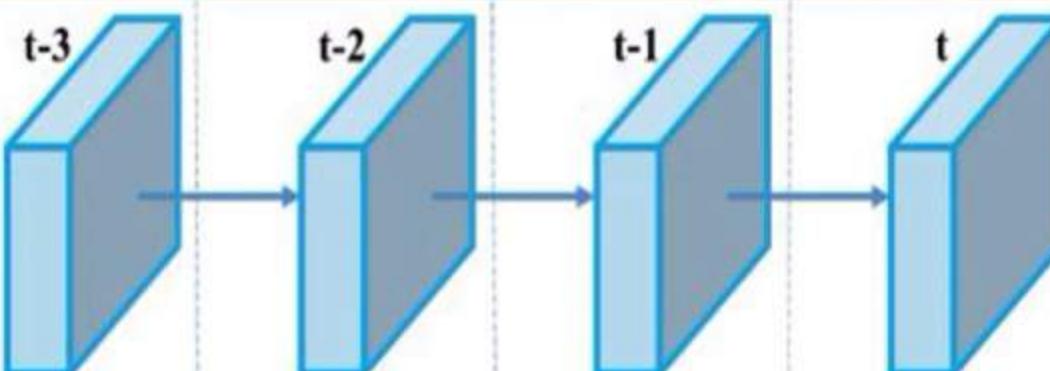
accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |  |
|--|---|---|--|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>calculate $BN^{(t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|--|---|---|--|

Gradient  
Accumulation



Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W

**CmBN – assume a batch contains four mini-batches**

|   |   |   |  |
|---|---|---|--|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|---|---|---|--|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

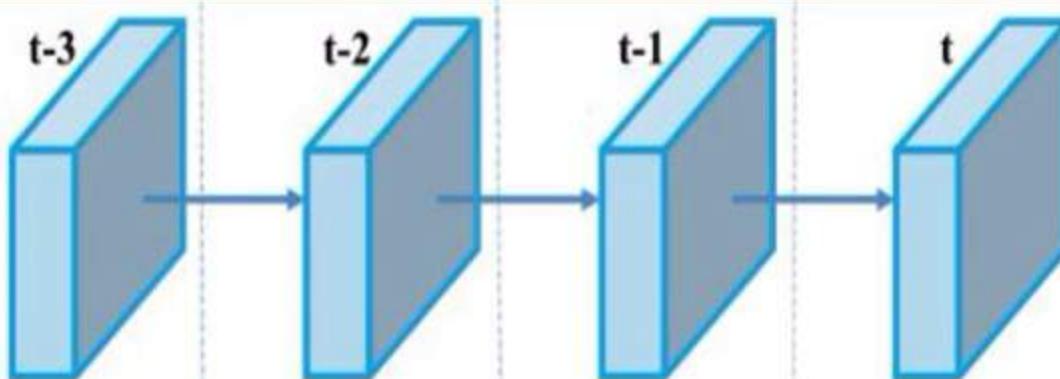
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

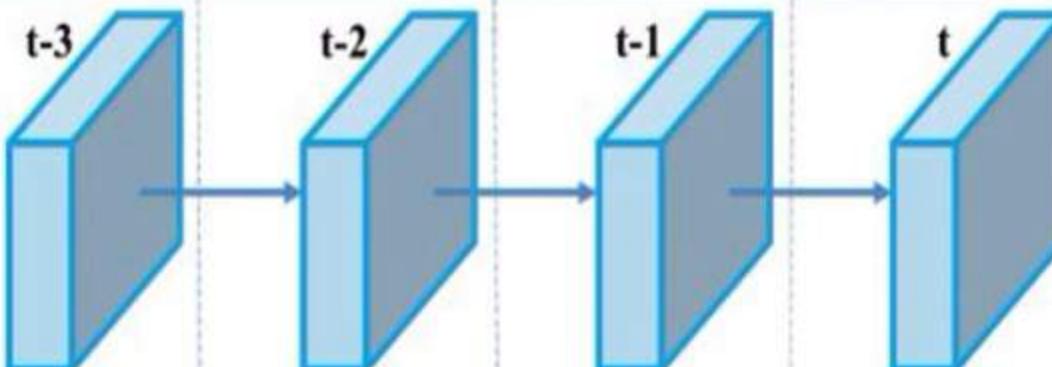
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation



Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W

**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

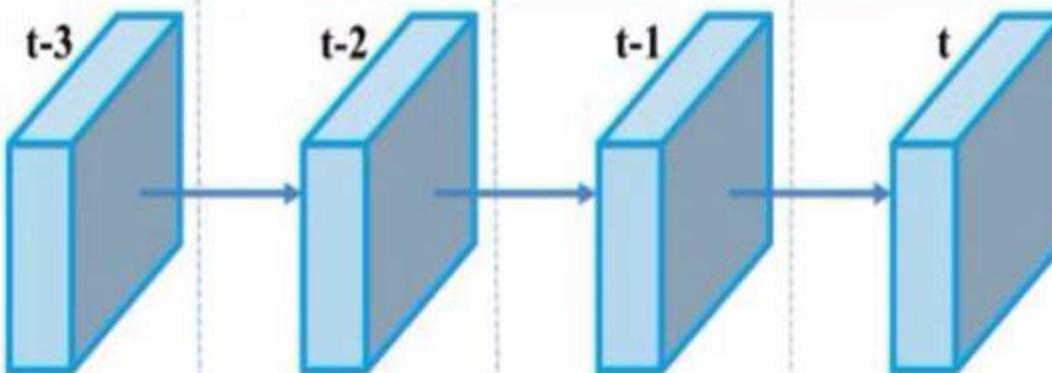
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

1

2

3

4

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

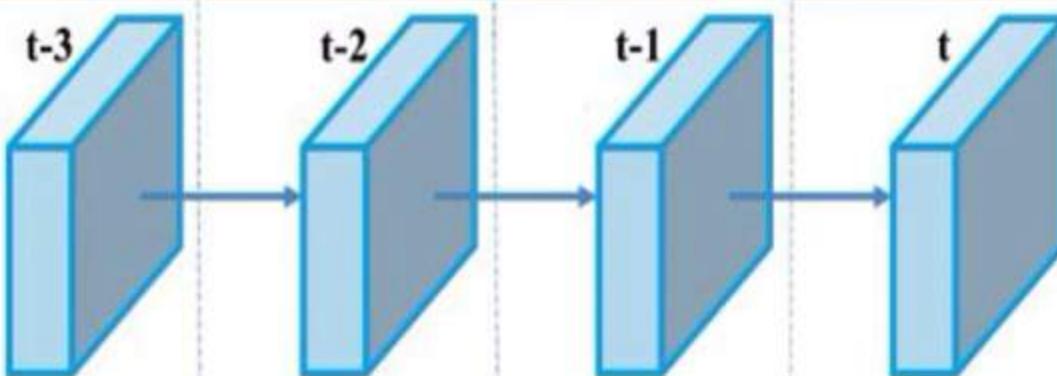
accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

1

2

3

4

# Cross Mini Batch Normalization

**BN [32]** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

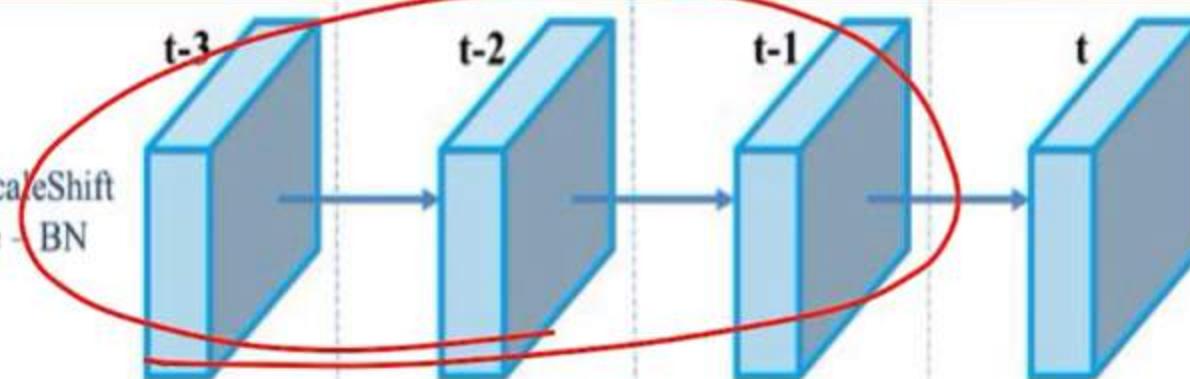
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

1

2

1, 2, 3

4

# Cross Mini Batch Normalization

**BN [32]** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

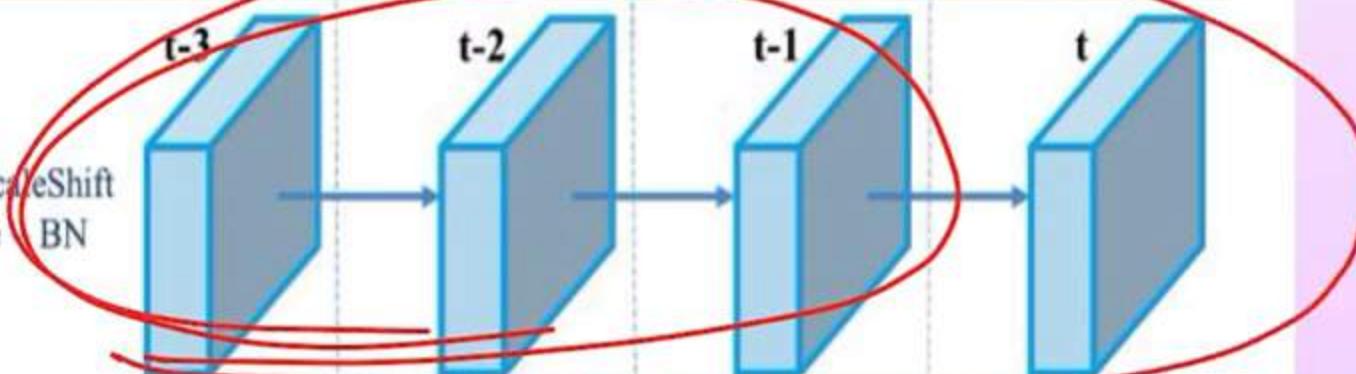
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update W, ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



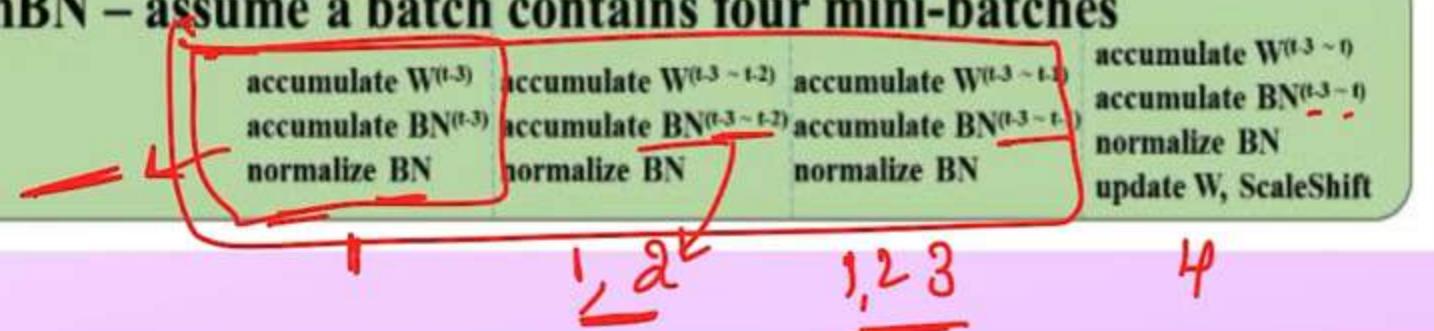
**CmBN** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update W, ScaleShift



# Cross Mini Batch Normalization

**BN [32]** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

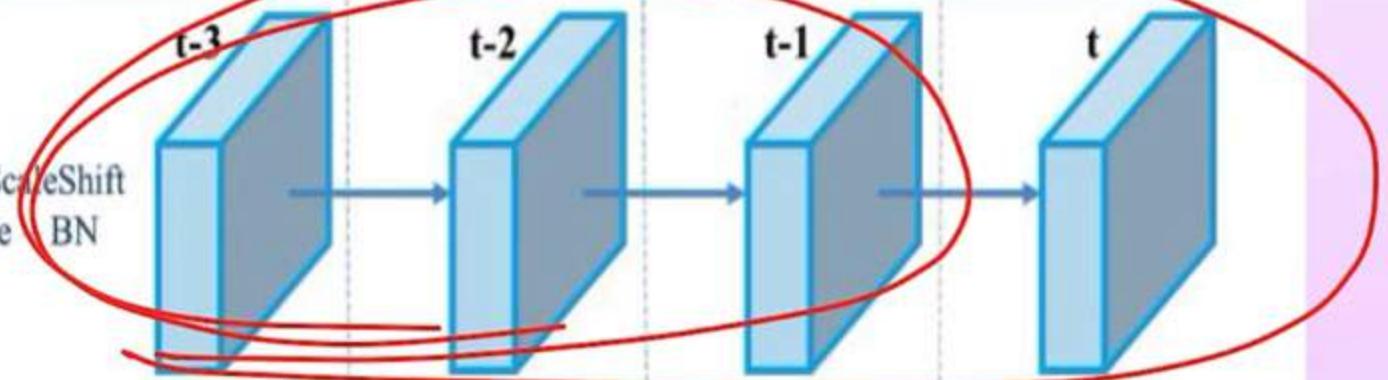
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update W, ScaleShift

Gradient Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



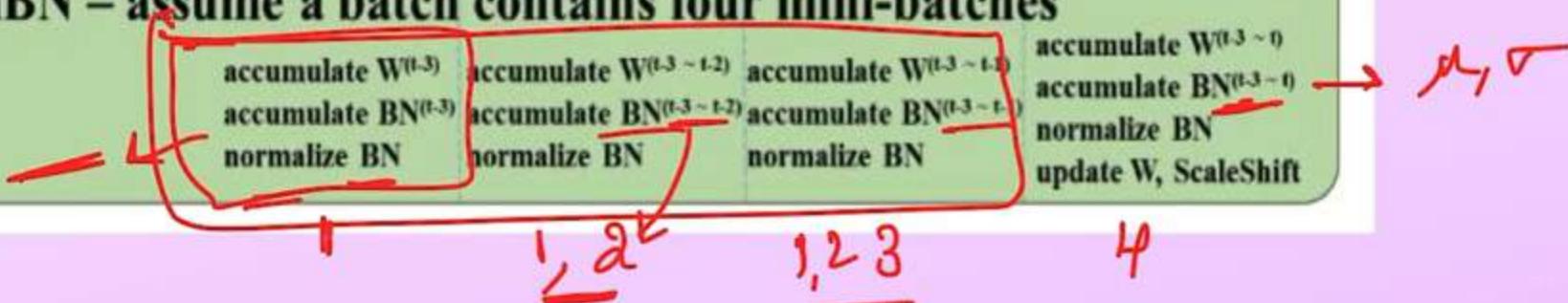
**CmBN** – assume a batch contains four mini-batches

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update W, ScaleShift



# Cross Mini Batch Normalization

**BN [32]** – assume a batch contains four mini-batches

accumulate  $W^{(l-3)}$   
calculate  $BN^{(l-3)}$   
normalize BN

accumulate  $W^{(l-3 \sim l-2)}$   
calculate  $BN^{(l-2)}$   
normalize BN

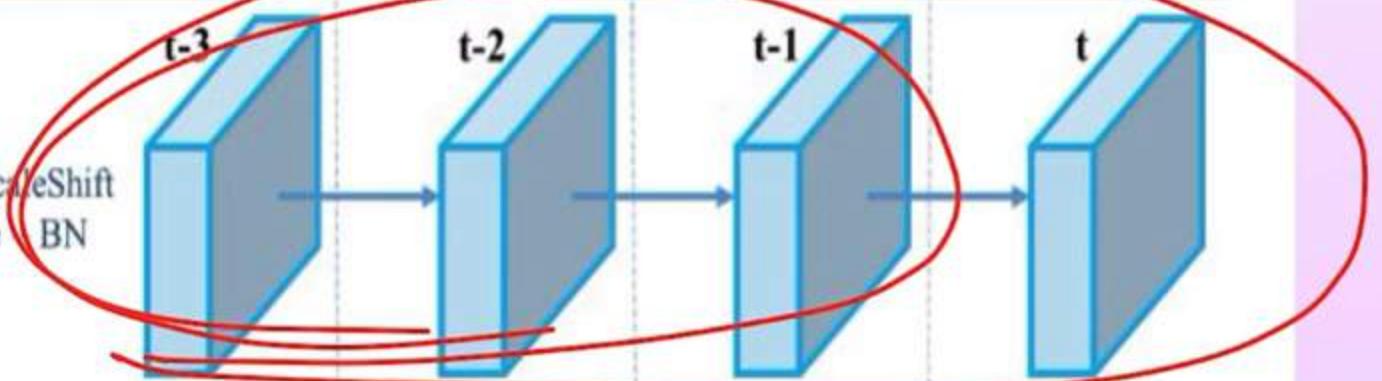
accumulate  $W^{(l-3 \sim l-1)}$   
calculate  $BN^{(l-1)}$   
normalize BN

accumulate  $W^{(l-3 \sim l)}$   
calculate  $BN^{(l)}$   
normalize BN  
update W, ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN** – assume a batch contains four mini-batches

accumulate  $W^{(l-3)}$   
accumulate  $BN^{(l-3)}$   
normalize BN

accumulate  $W^{(l-3 \sim l-2)}$   
accumulate  $BN^{(l-3 \sim l-2)}$   
normalize BN

accumulate  $W^{(l-3 \sim l-1)}$   
accumulate  $BN^{(l-3 \sim l-1)}$   
normalize BN

accumulate  $W^{(l-3 \sim l)}$   
accumulate  $BN^{(l-3 \sim l)}$   
normalize BN  
update W, ScaleShift

1, 2

1, 2, 3

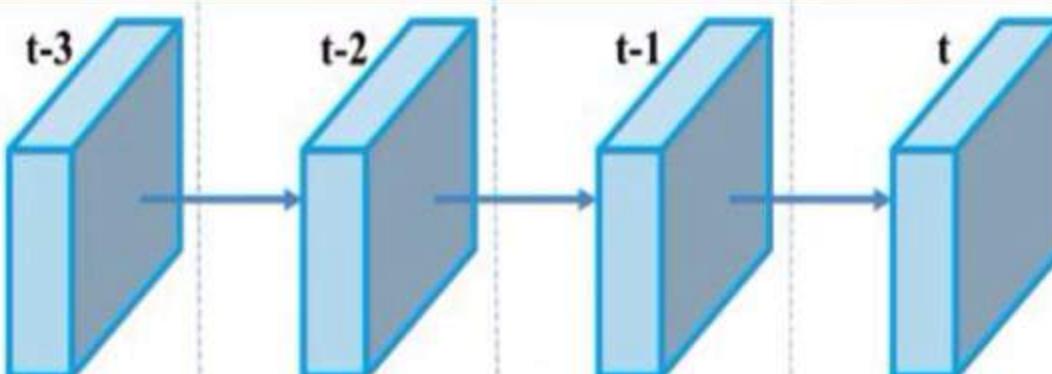
4 → 1, 2, 3, 4

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

|  |   |   |  |
|--|---|---|--|
| accumulate $W^{(t-3)}$<br>calculate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>calculate $BN^{(t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>calculate $BN^{(t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>calculate $BN^{(t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|--|---|---|--|

Gradient  
Accumulation



**CmBN – assume a batch contains four mini-batches**

|   |   |   |  |
|---|---|---|--|
| accumulate $W^{(t-3)}$<br>accumulate $BN^{(t-3)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-2)}$<br>accumulate $BN^{(t-3 \sim t-2)}$<br>normalize BN | accumulate $W^{(t-3 \sim t-1)}$<br>accumulate $BN^{(t-3 \sim t-1)}$<br>normalize BN | accumulate $W^{(t-3 \sim t)}$<br>accumulate $BN^{(t-3 \sim t)}$<br>normalize BN<br>update $W$ , ScaleShift |
|---|---|---|--|

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

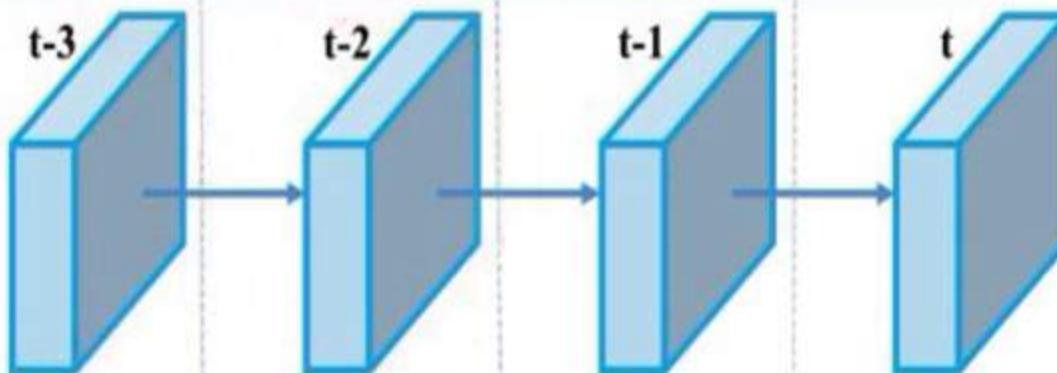
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

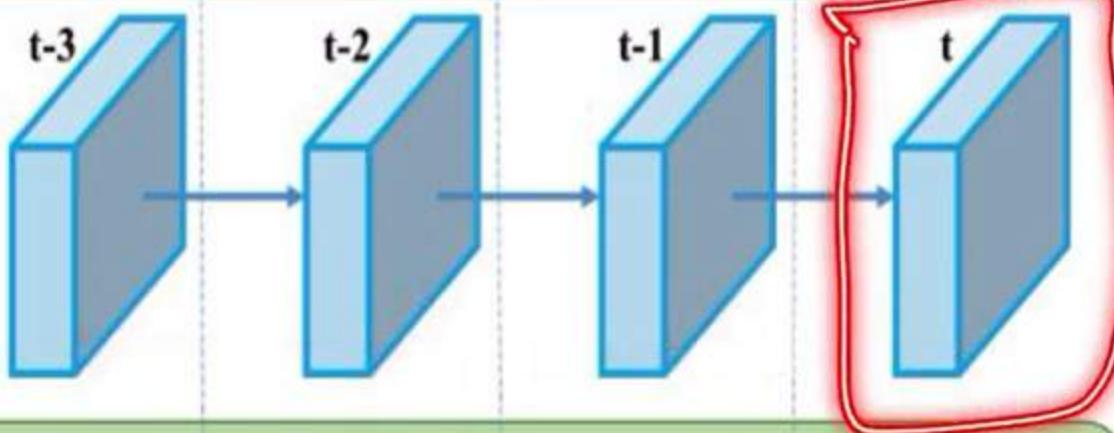
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Cross Mini Batch Normalization

**BN [32] – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
calculate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
calculate  $BN^{(t-2)}$   
normalize BN

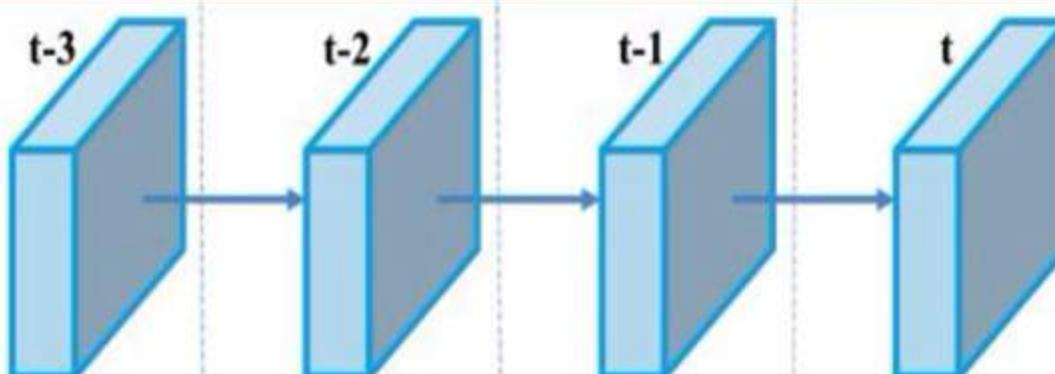
accumulate  $W^{(t-3 \sim t-1)}$   
calculate  $BN^{(t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
calculate  $BN^{(t)}$   
normalize BN  
update  $W$ , ScaleShift

Gradient  
Accumulation

Lets:

Bias, scale – ScaleShift  
Mean, variance – BN  
Weights – W



**CmBN – assume a batch contains four mini-batches**

accumulate  $W^{(t-3)}$   
accumulate  $BN^{(t-3)}$   
normalize BN

accumulate  $W^{(t-3 \sim t-2)}$   
accumulate  $BN^{(t-3 \sim t-2)}$   
normalize BN

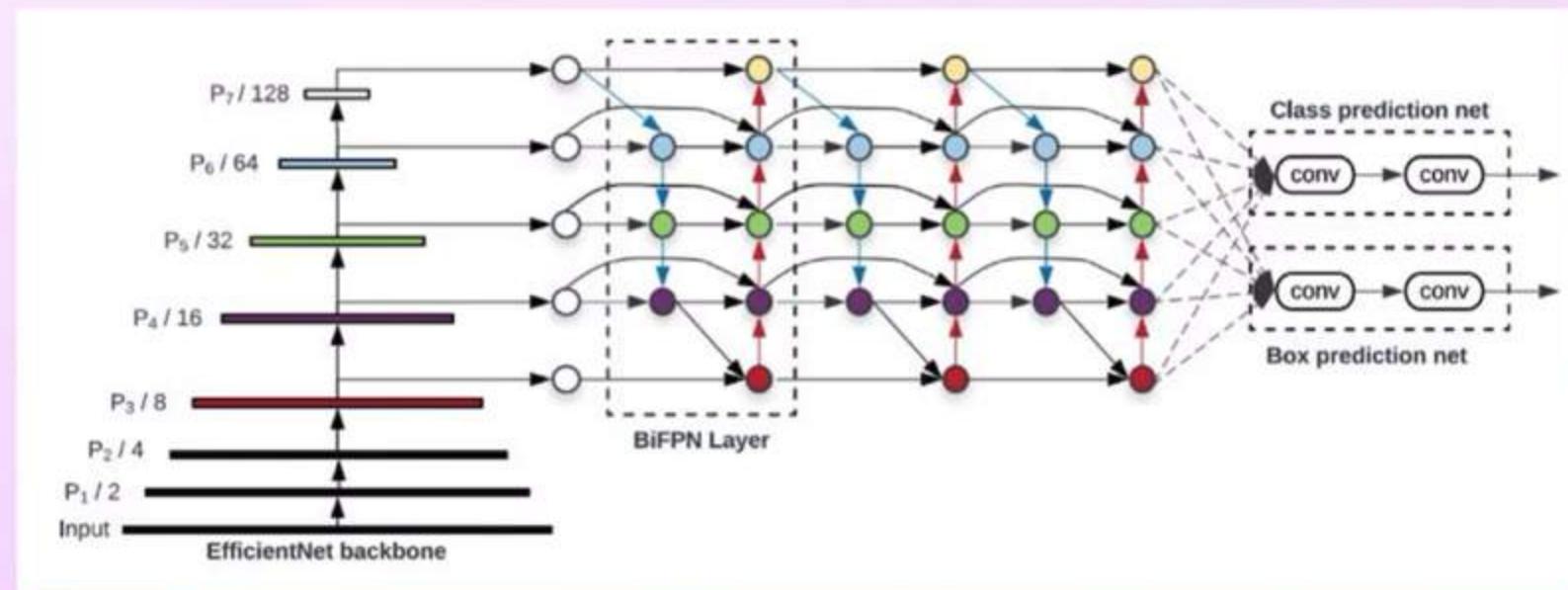
accumulate  $W^{(t-3 \sim t-1)}$   
accumulate  $BN^{(t-3 \sim t-1)}$   
normalize BN

accumulate  $W^{(t-3 \sim t)}$   
accumulate  $BN^{(t-3 \sim t)}$   
normalize BN  
update  $W$ , ScaleShift

# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

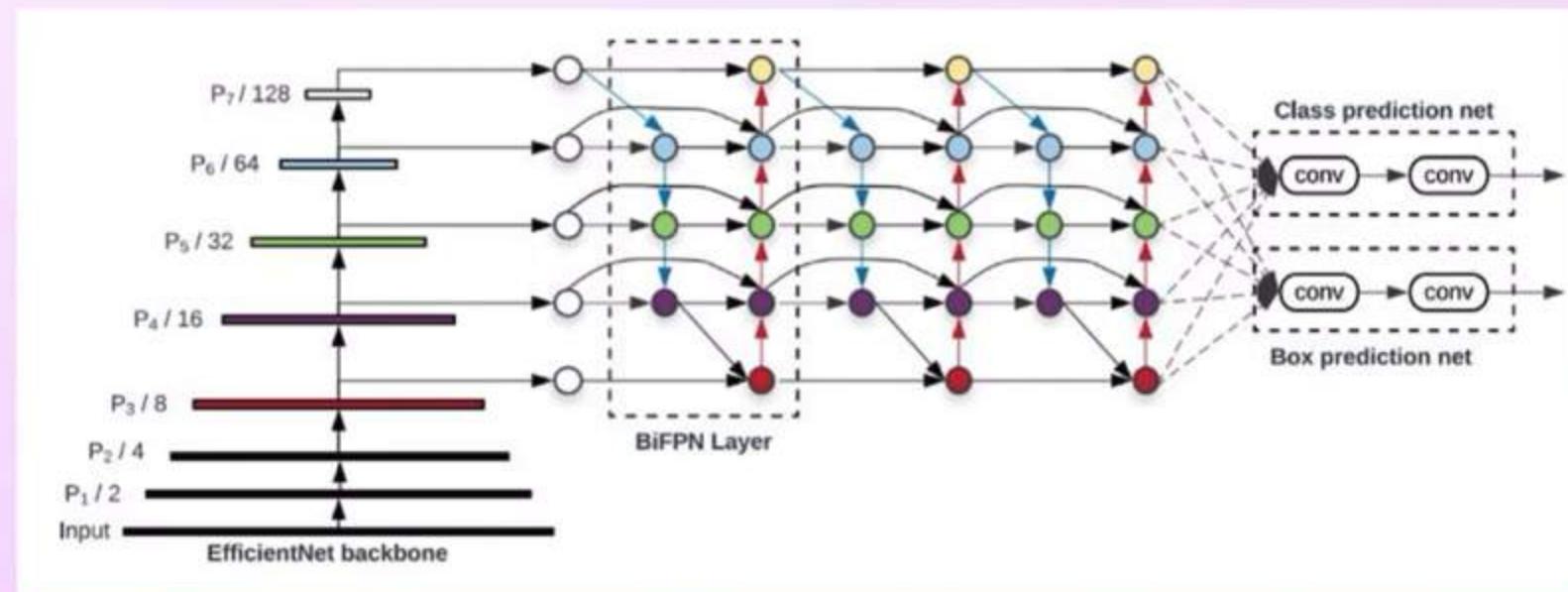
- . EfficientDet →



# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

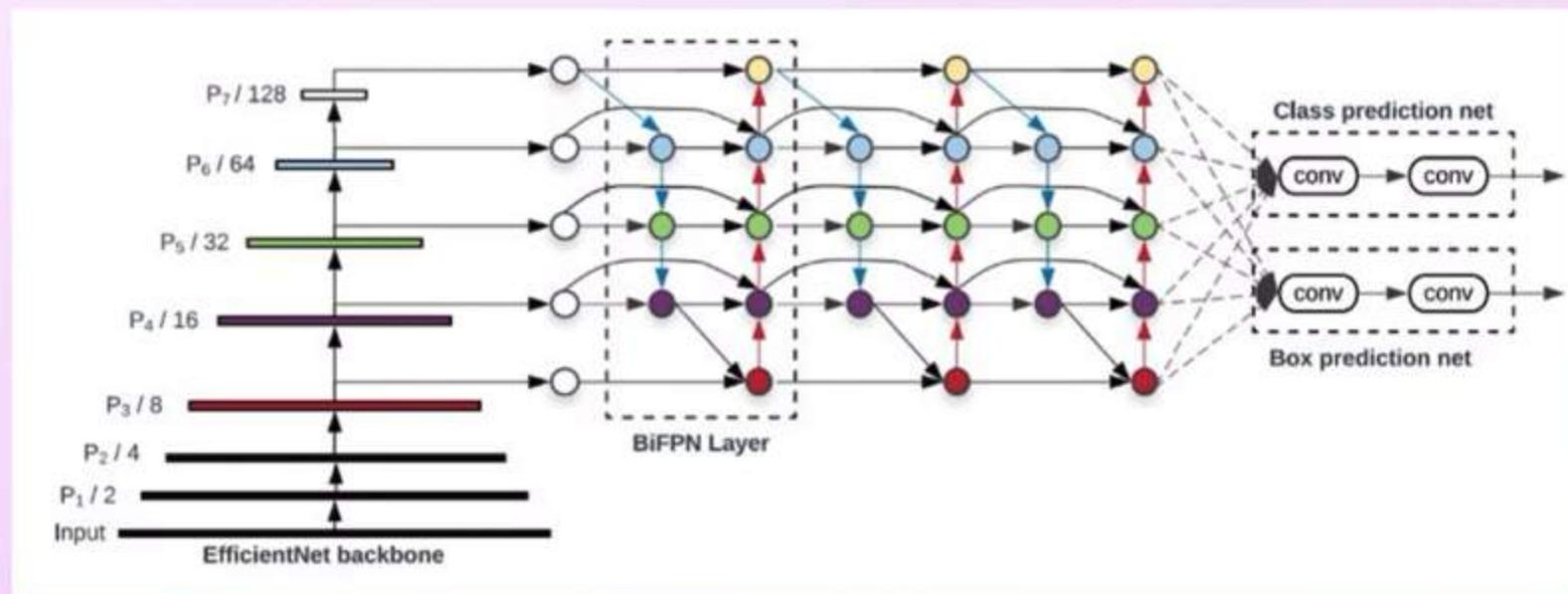
- . EfficientDet →



# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

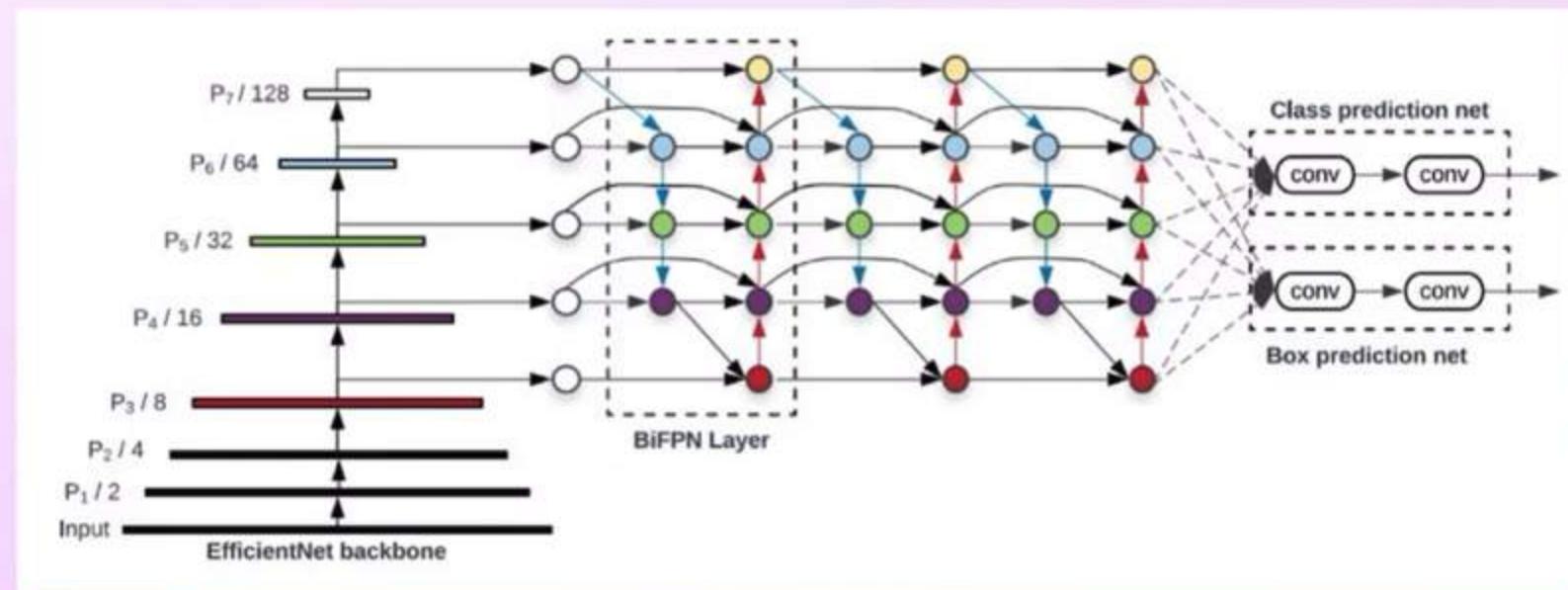
- . EfficientDet →



# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

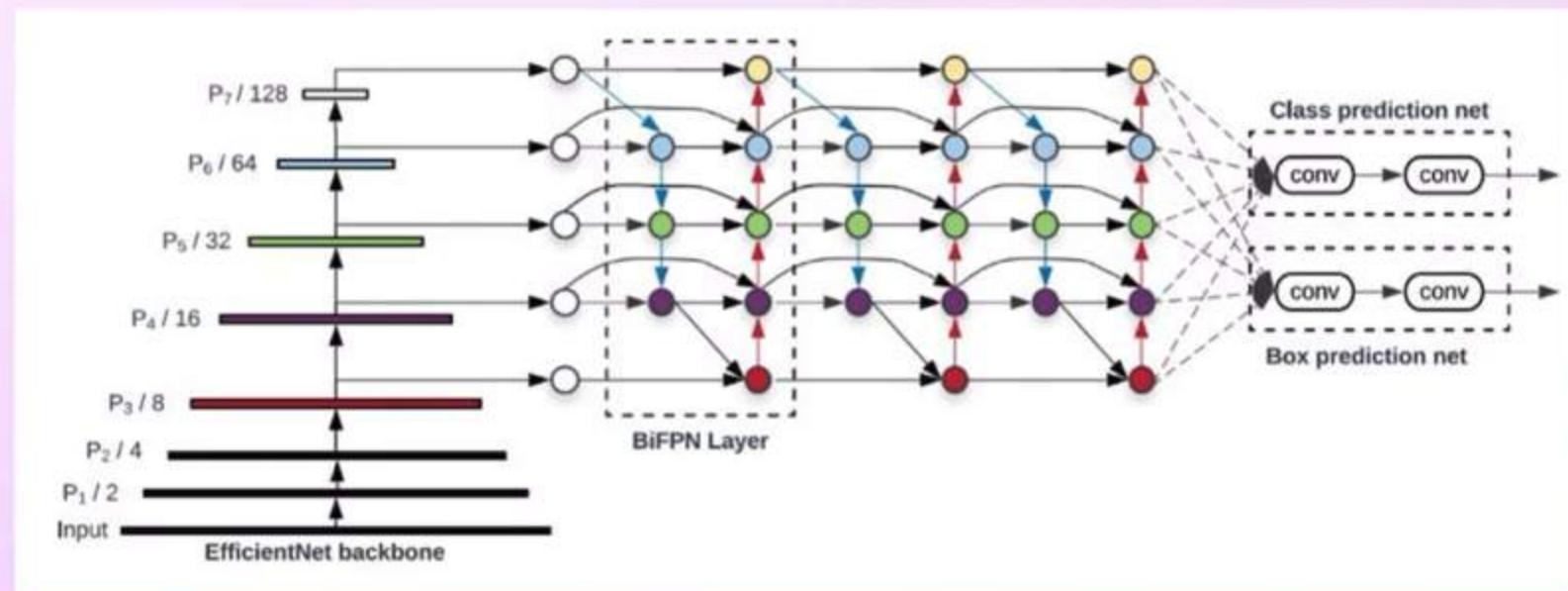
. EfficientDet →



# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

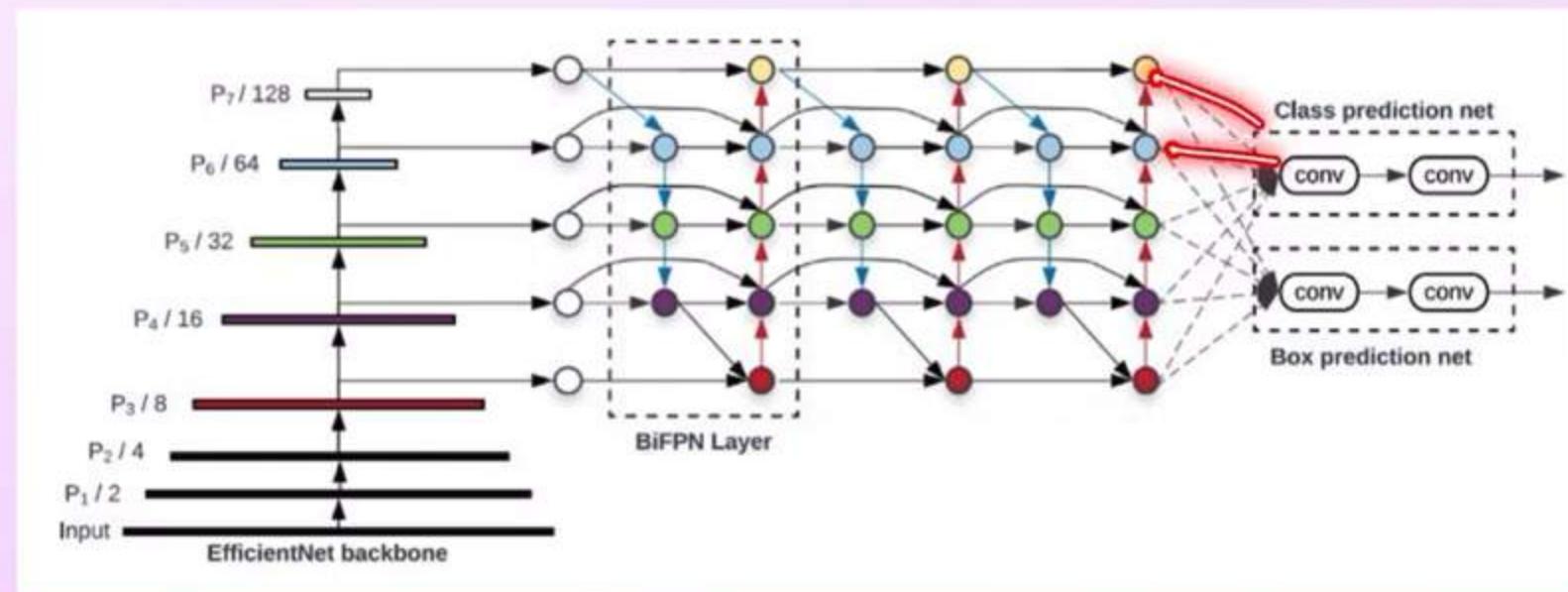
- . EfficientDet →



# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

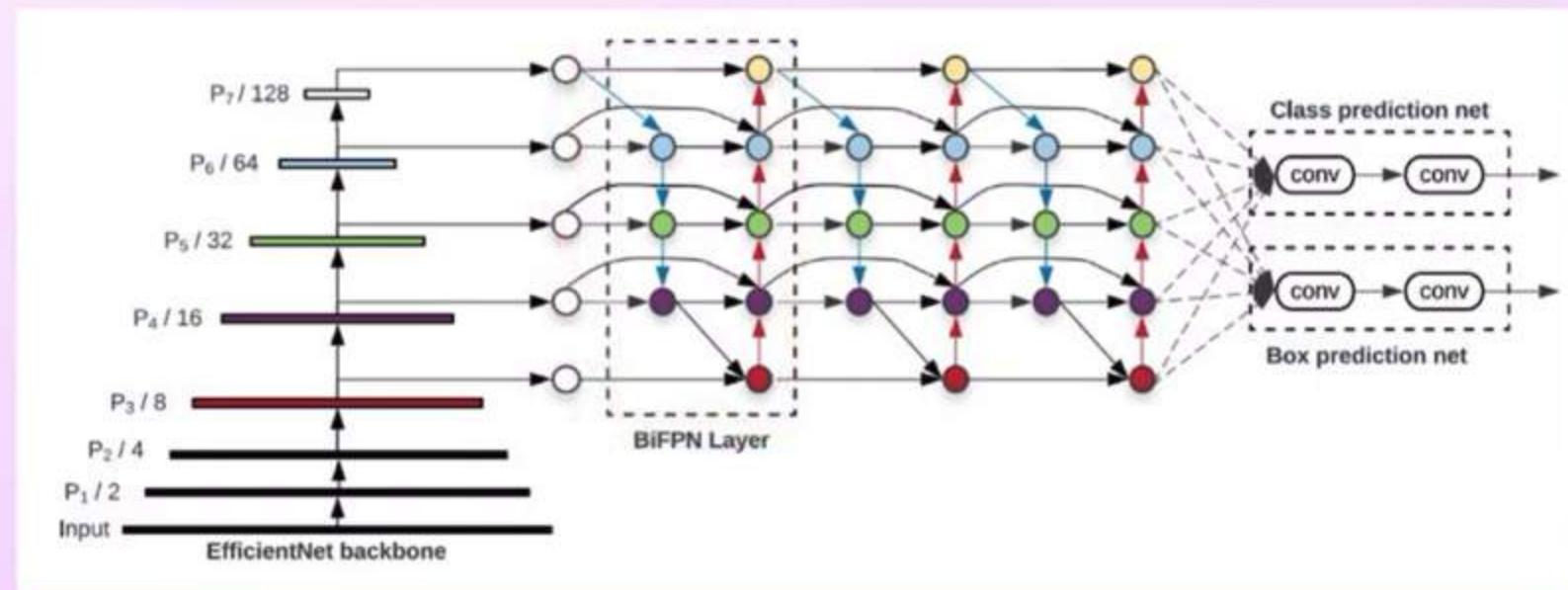
- . EfficientDet →



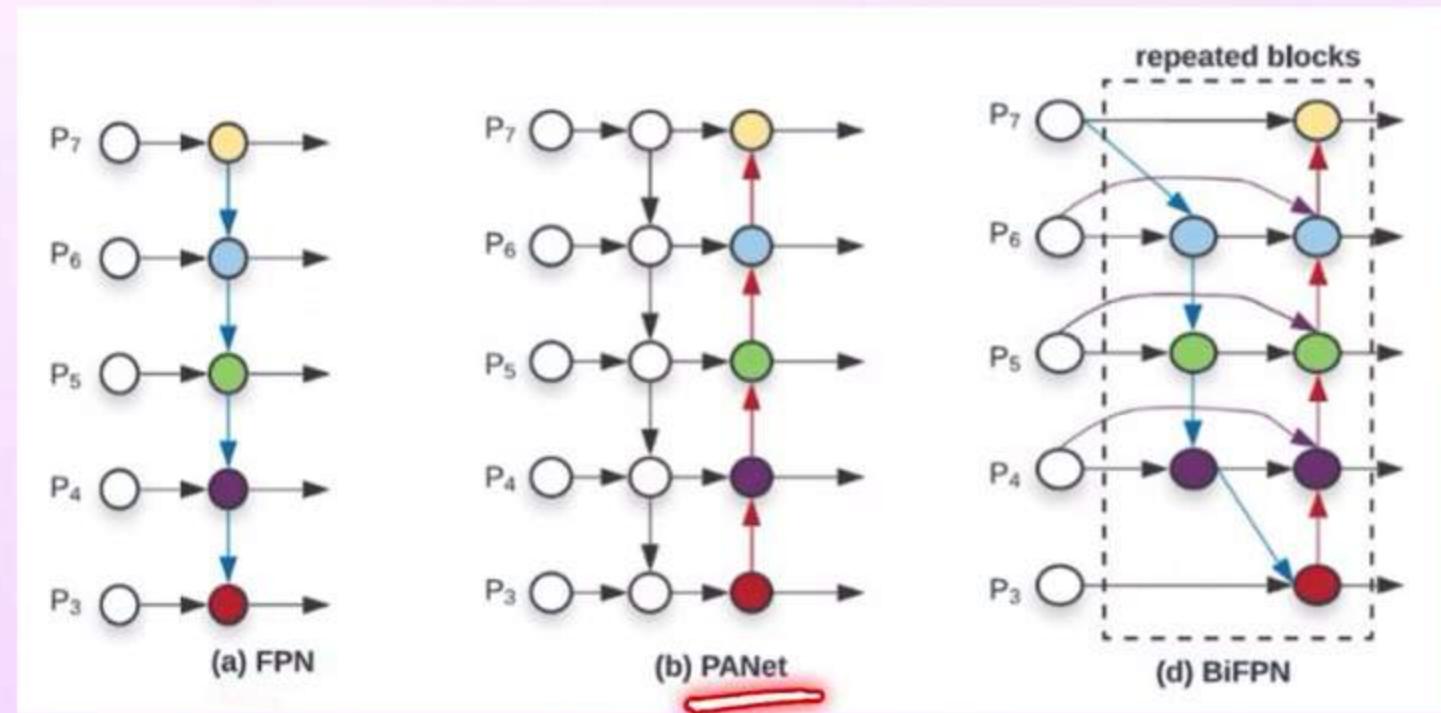
# Multi-Input Weighted Residual Connections

- . Idea from EfficientDet Paper
- . Stage: Neck

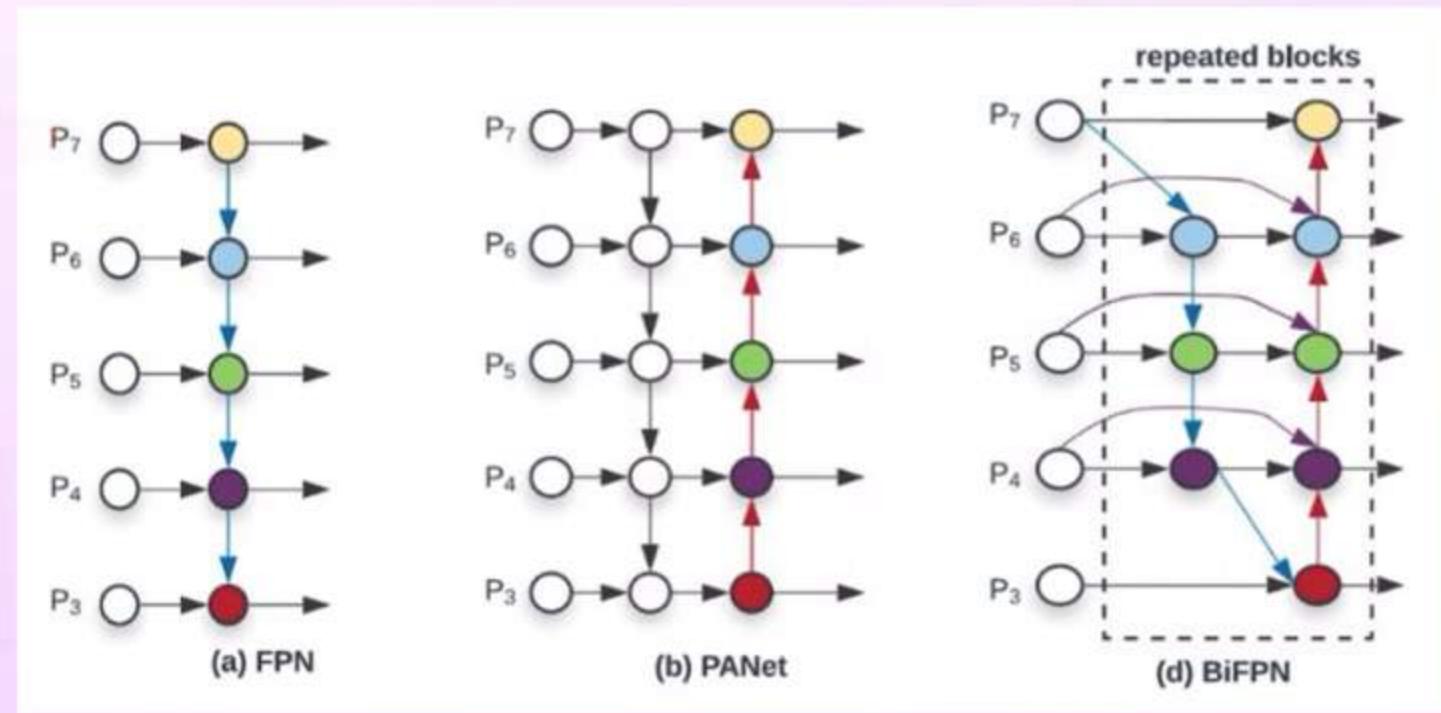
- . EfficientDet →



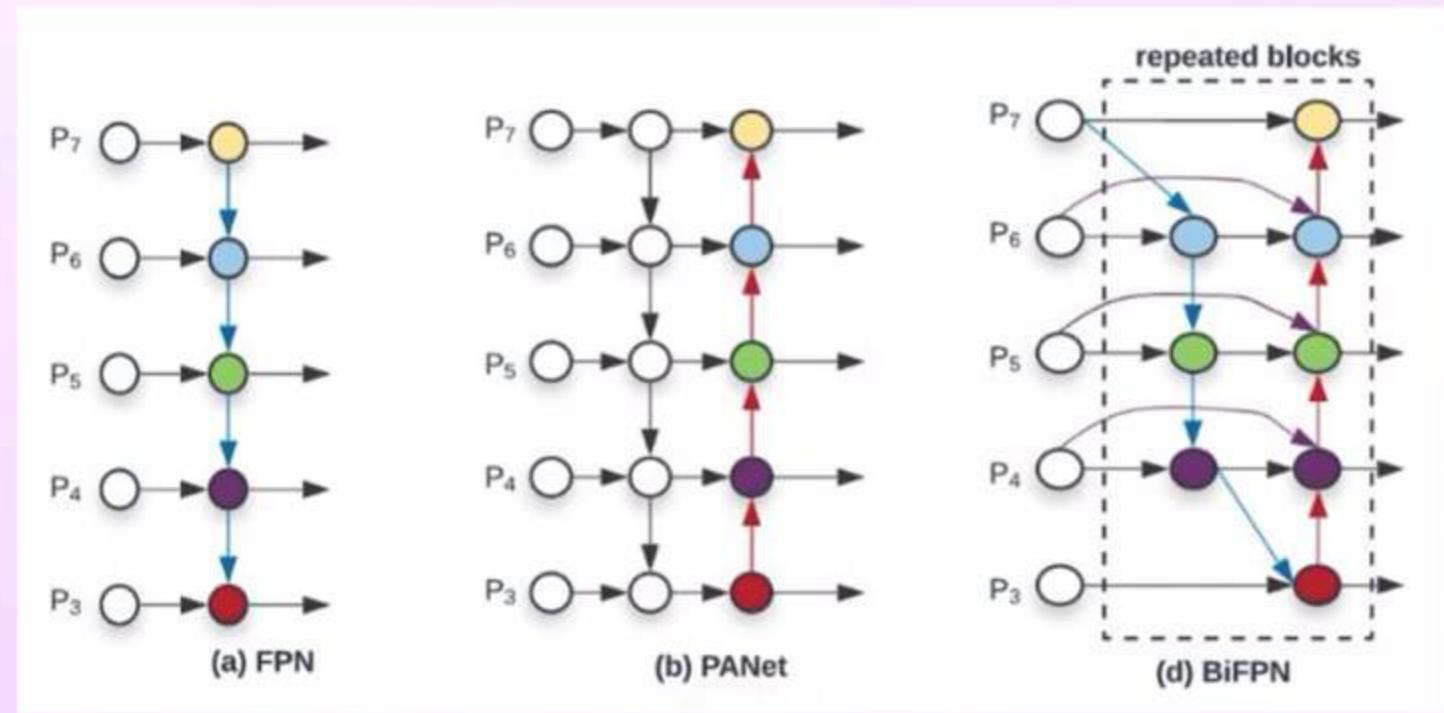
# Multi-Input Weighted Residual Connections



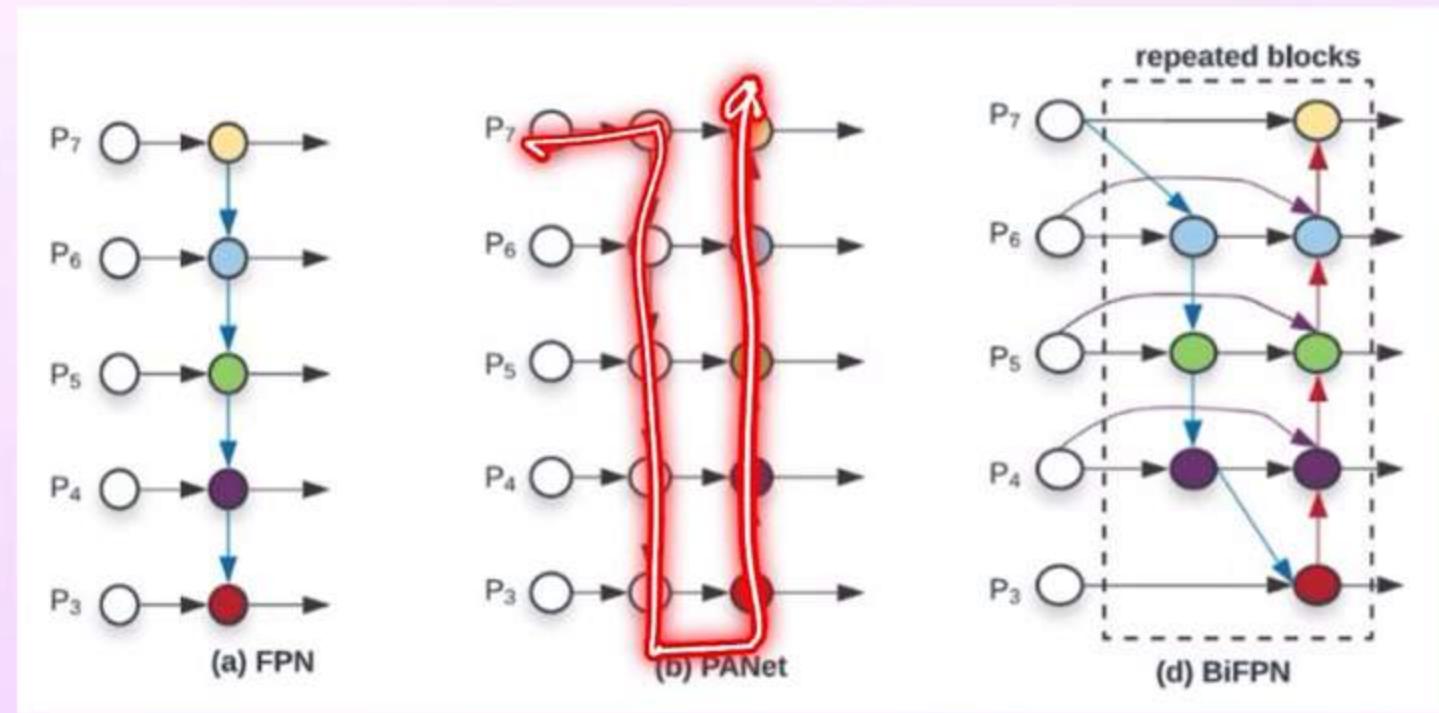
# Multi-Input Weighted Residual Connections



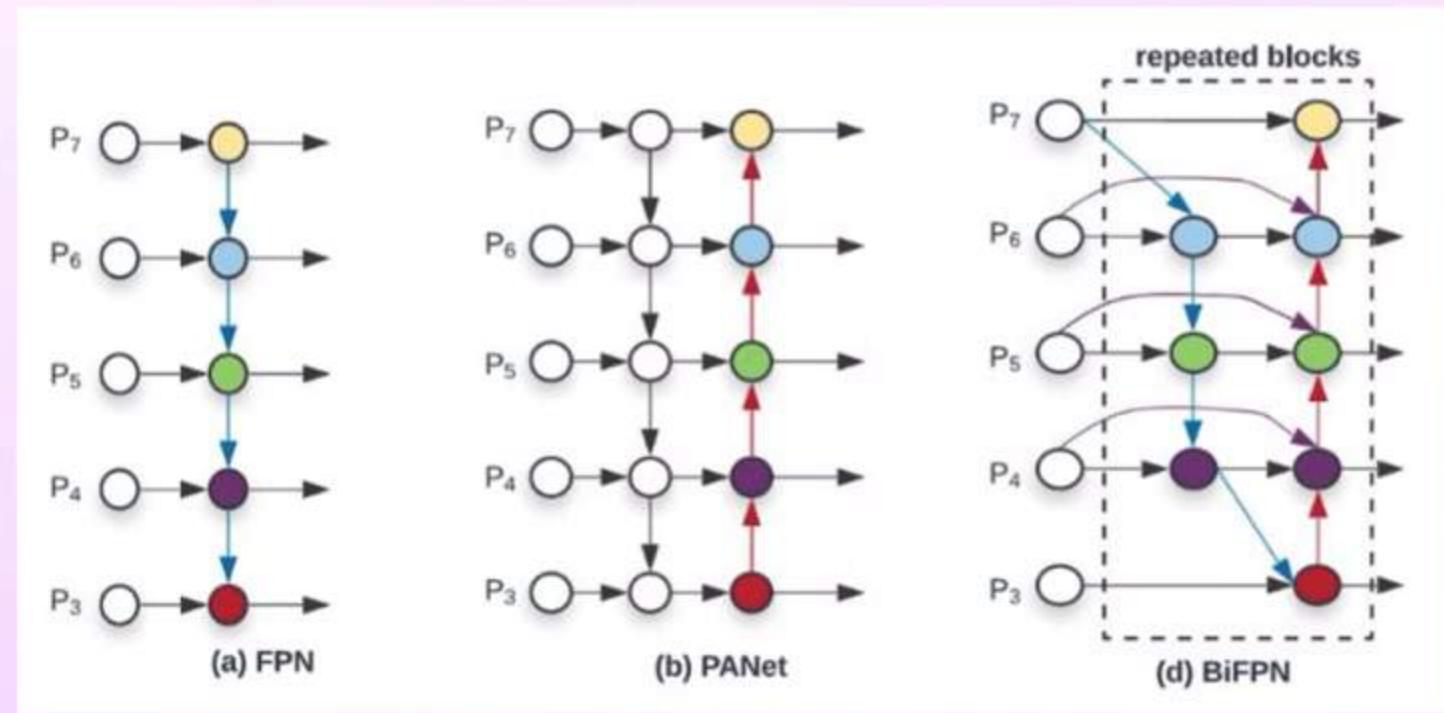
# Multi-Input Weighted Residual Connections



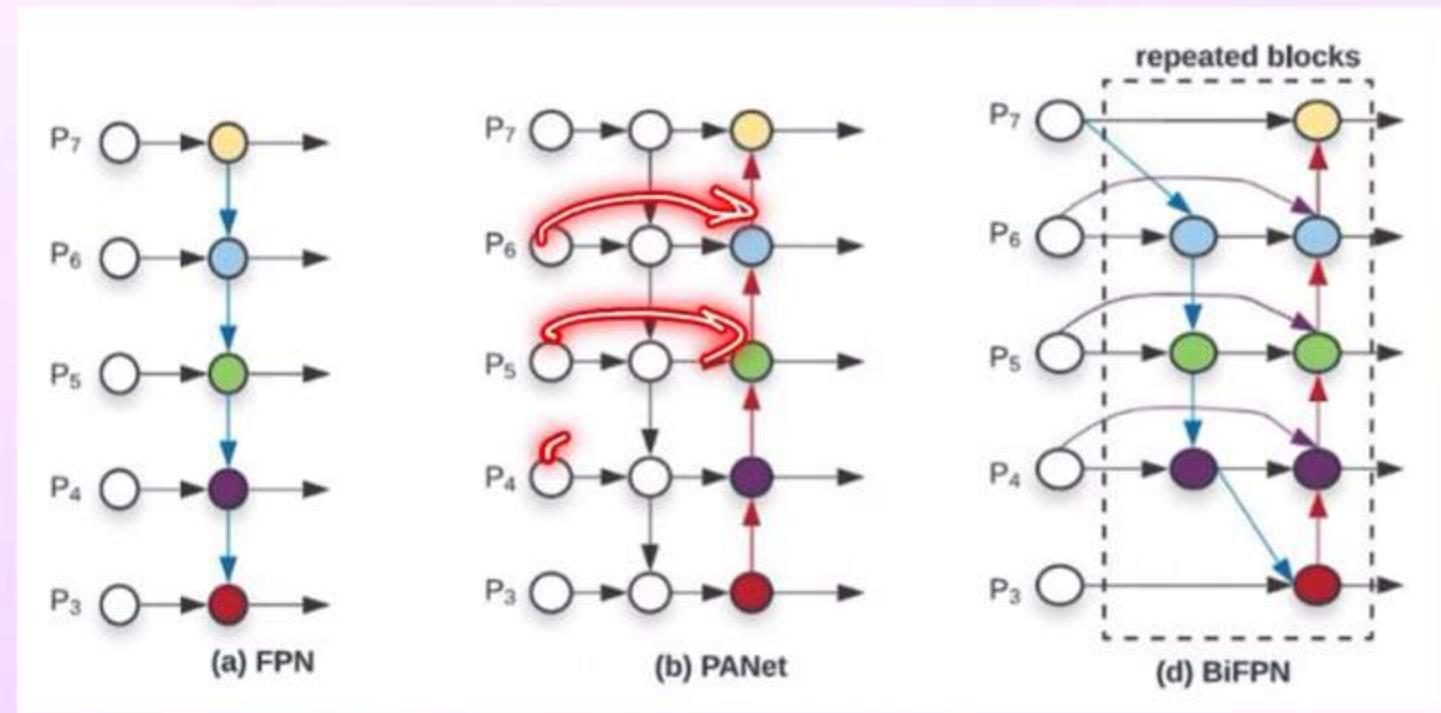
# Multi-Input Weighted Residual Connections



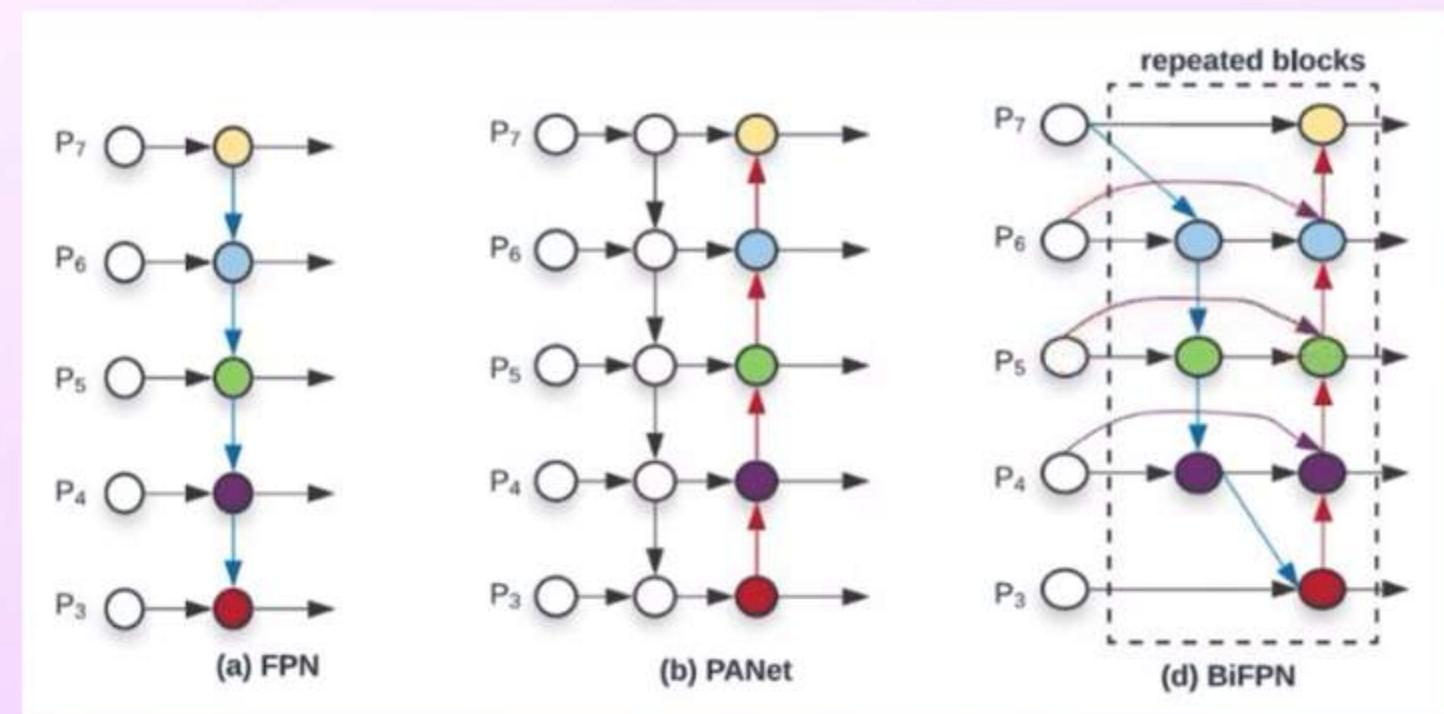
# Multi-Input Weighted Residual Connections



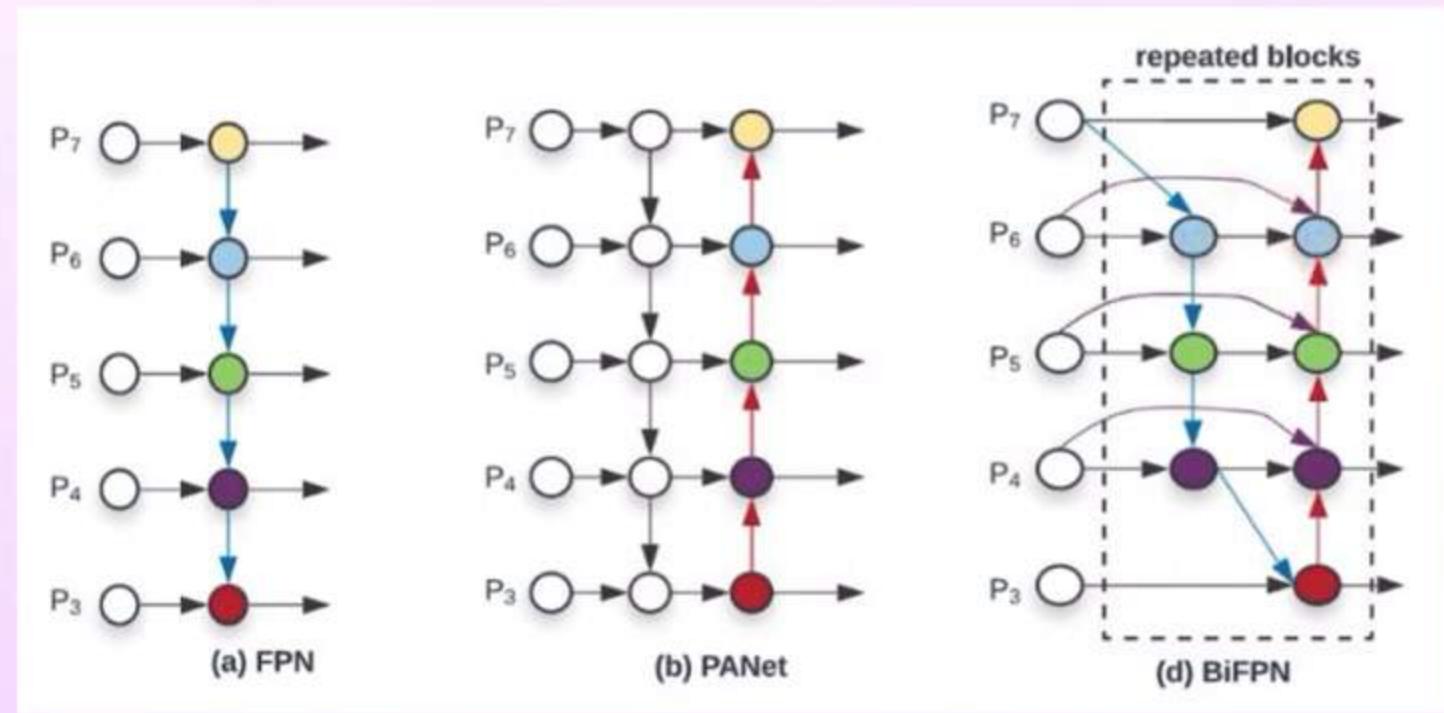
# Multi-Input Weighted Residual Connections



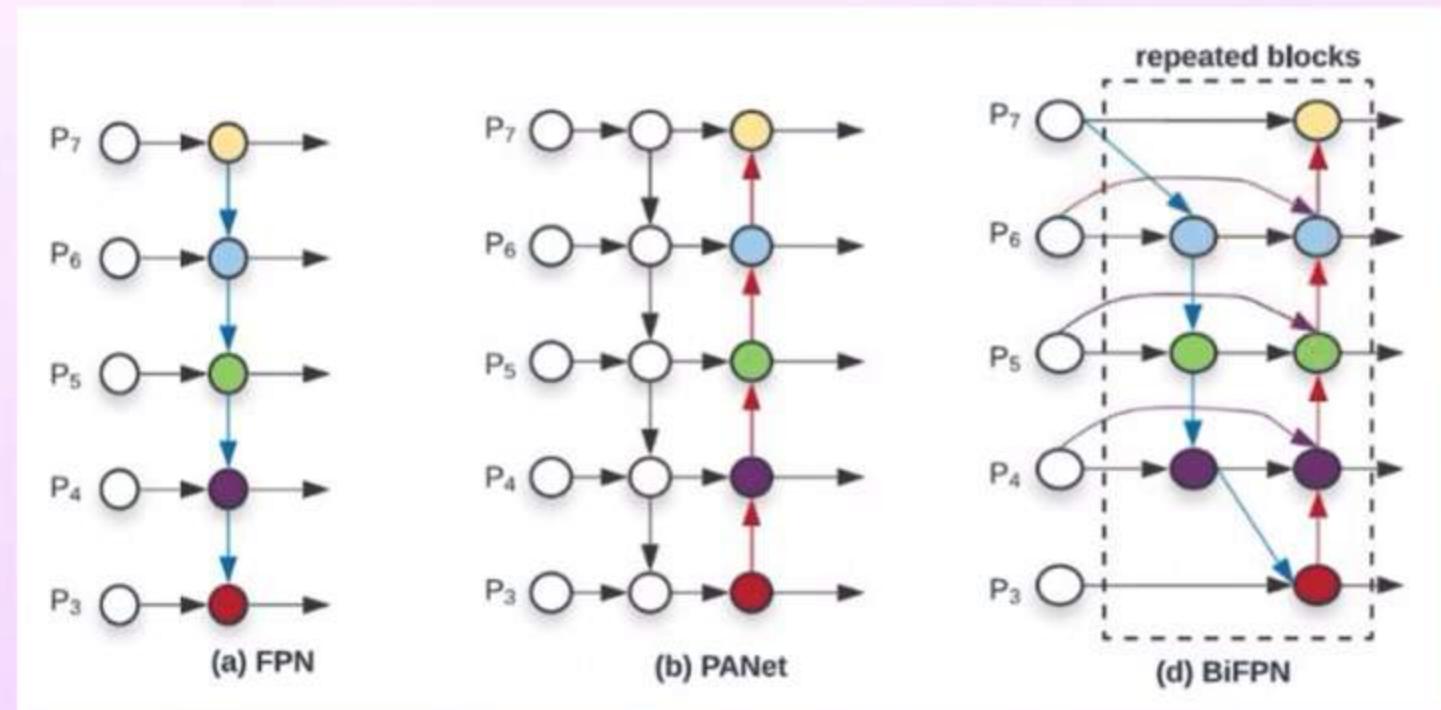
# Multi-Input Weighted Residual Connections



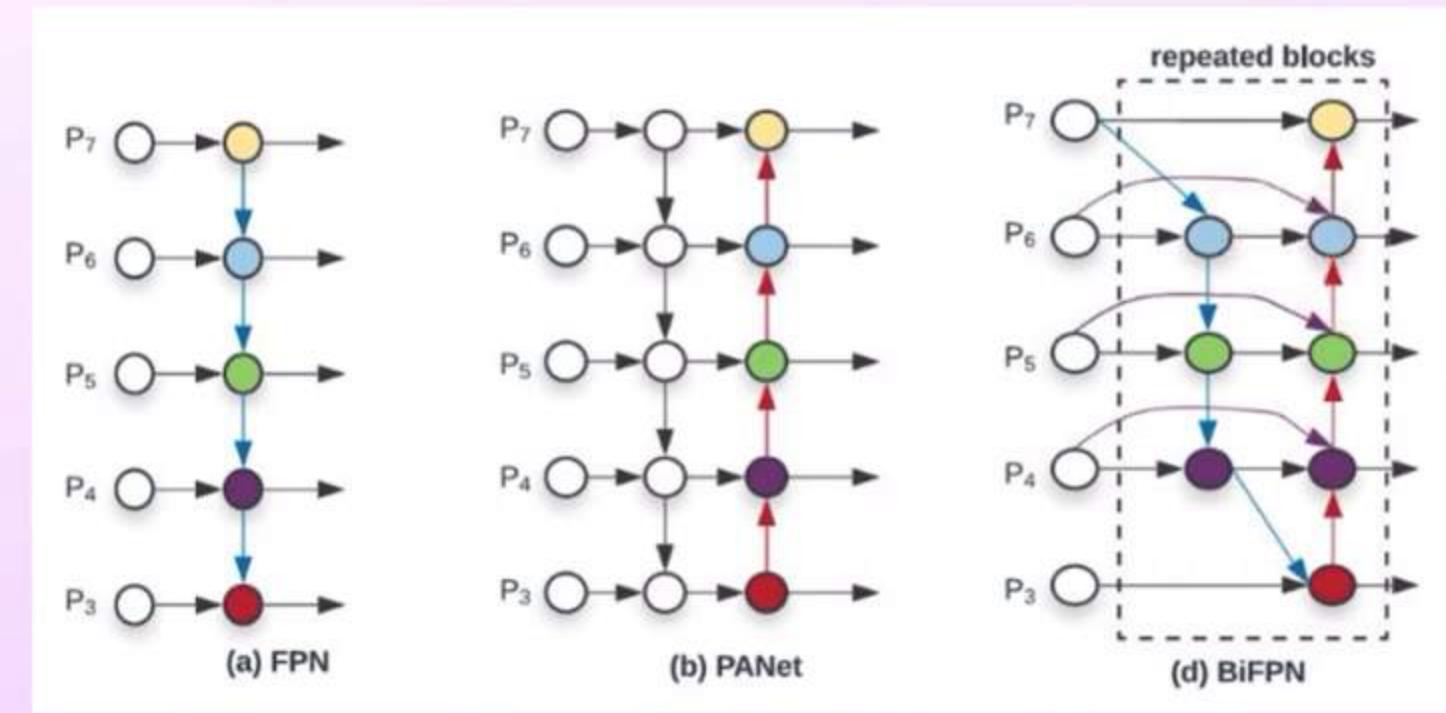
# Multi-Input Weighted Residual Connections



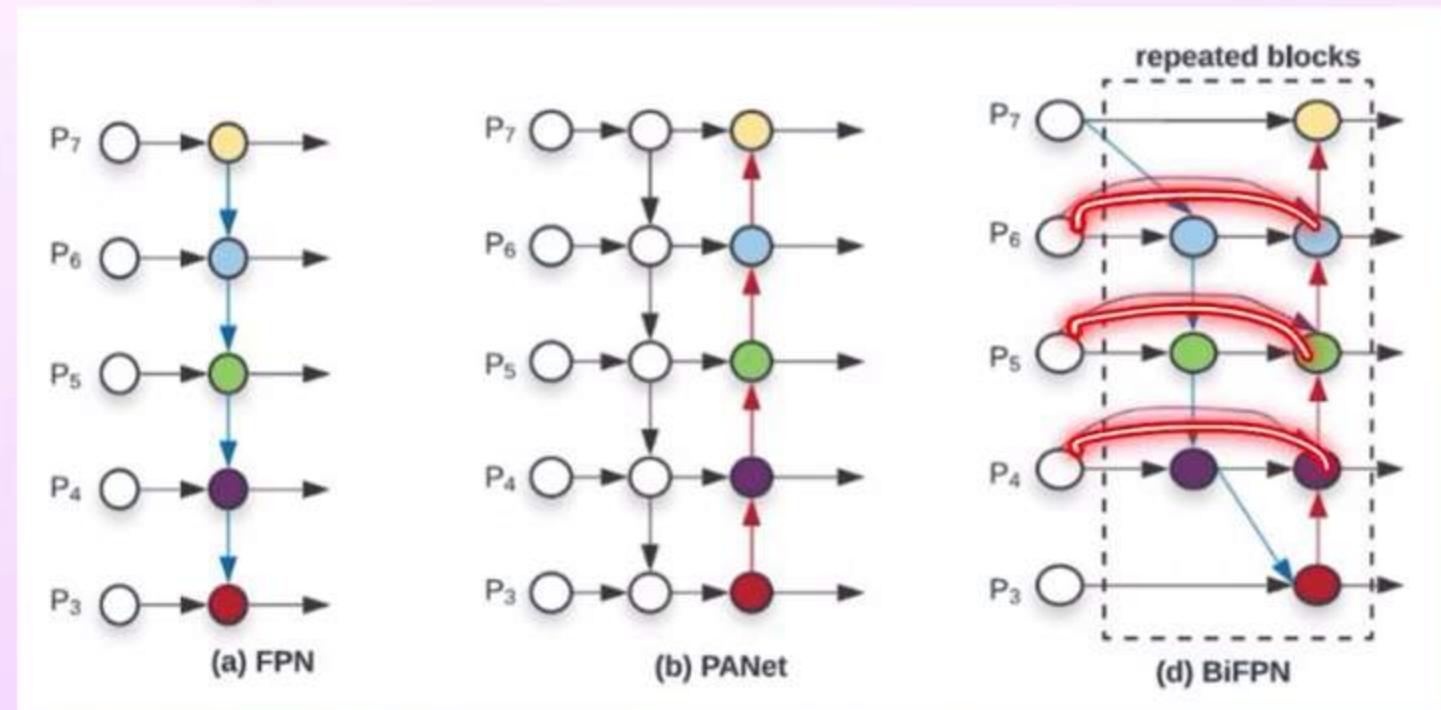
# Multi-Input Weighted Residual Connections



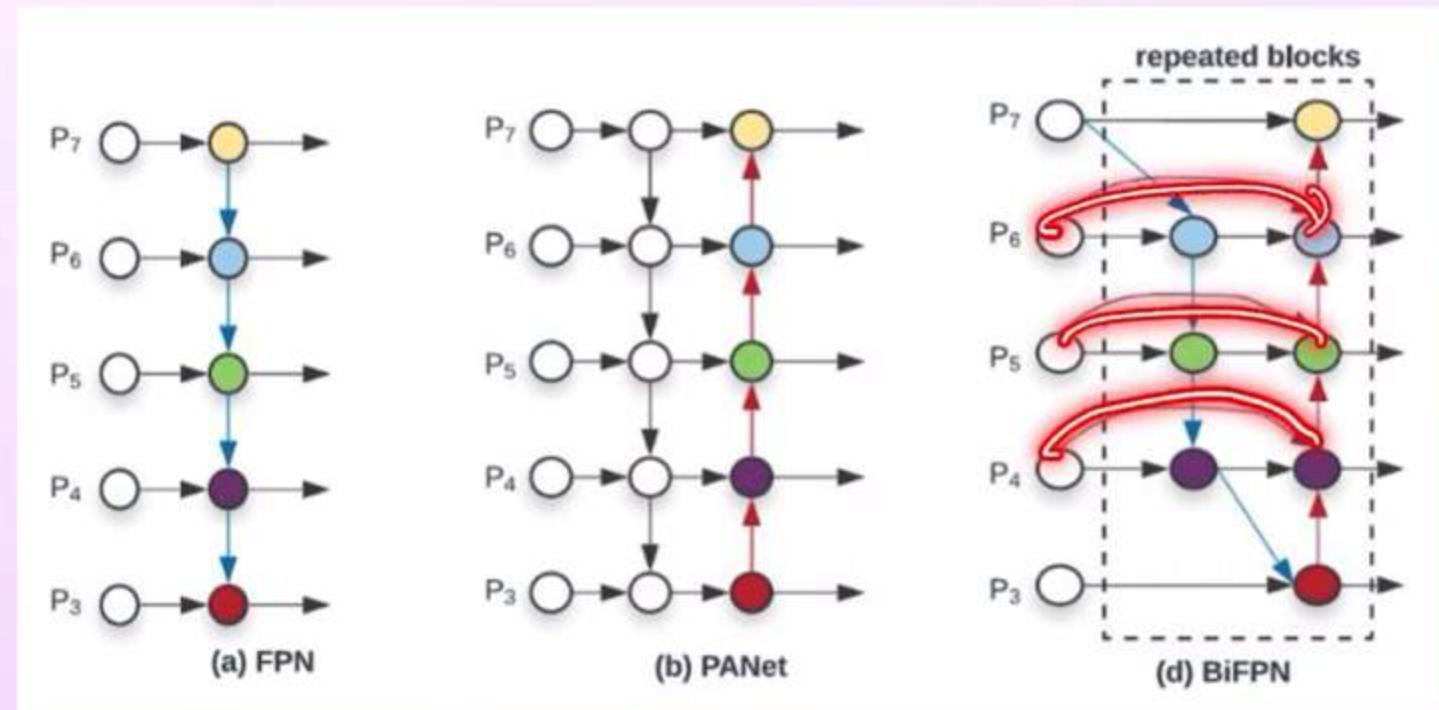
# Multi-Input Weighted Residual Connections



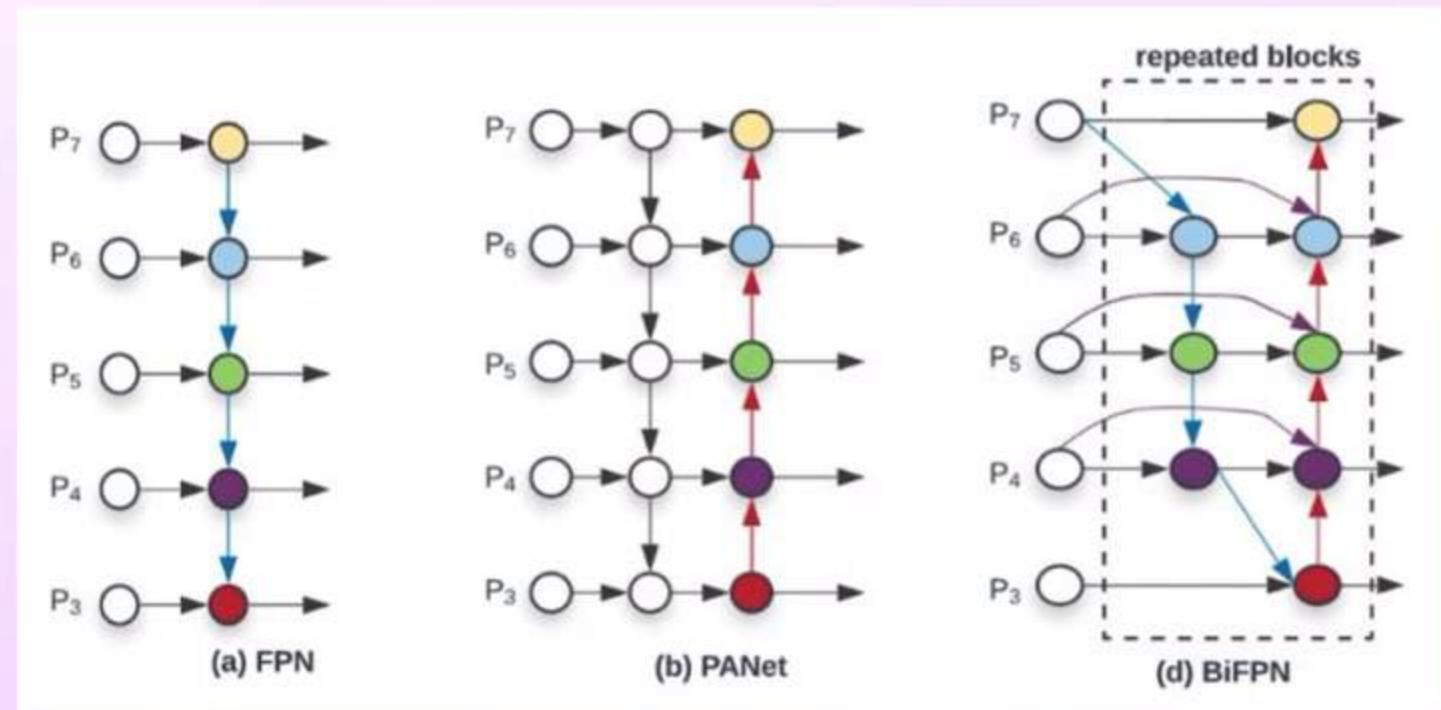
# Multi-Input Weighted Residual Connections



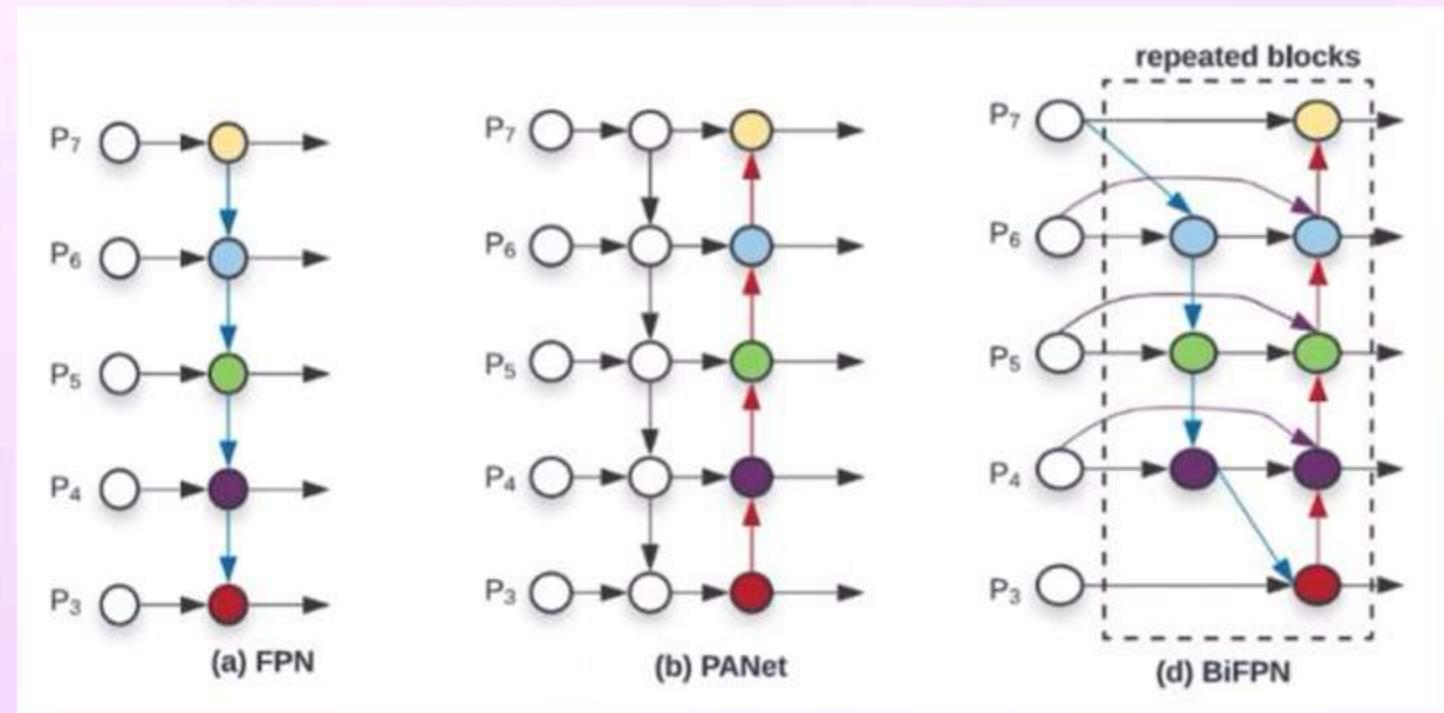
# Multi-Input Weighted Residual Connections



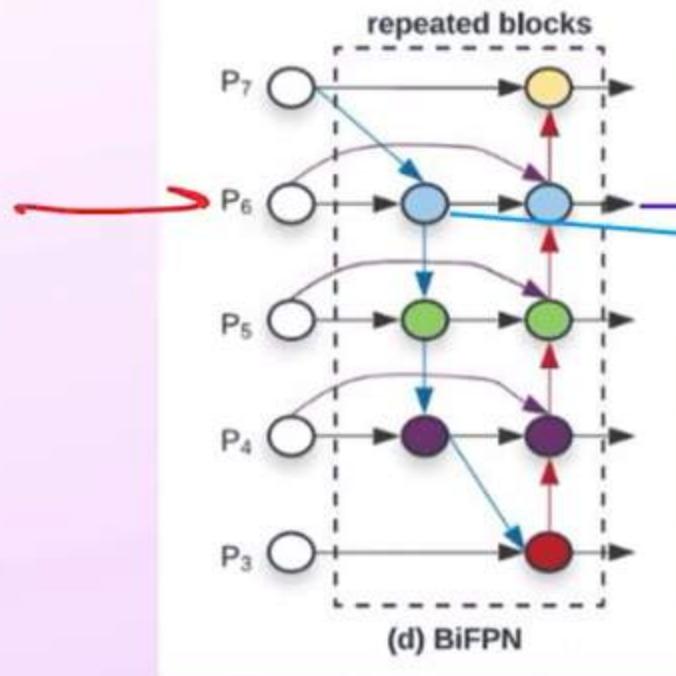
# Multi-Input Weighted Residual Connections



# Multi-Input Weighted Residual Connections



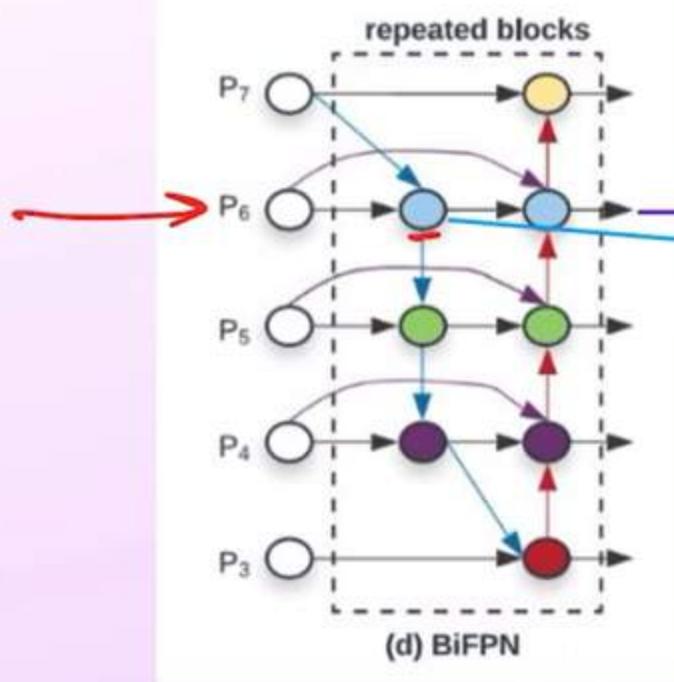
# Multi-Input Weighted Residual Connections



$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$

$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

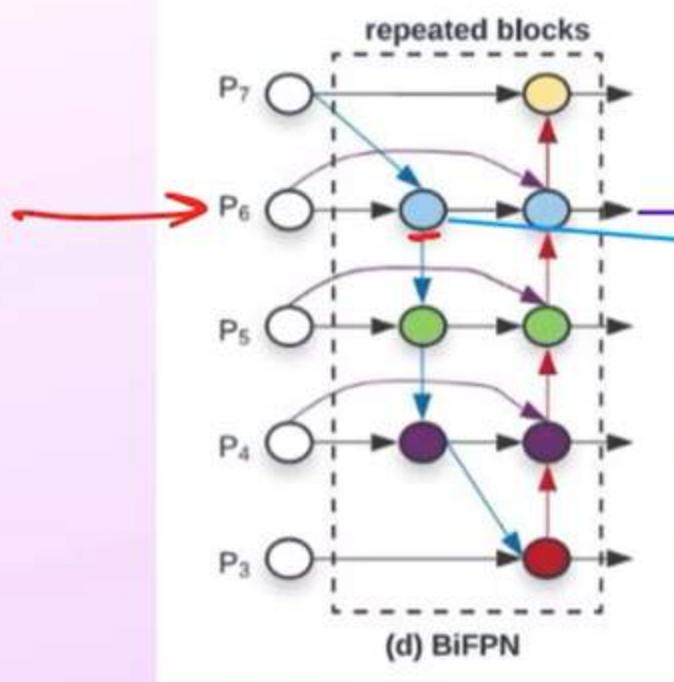
# Multi-Input Weighted Residual Connections



$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$

$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

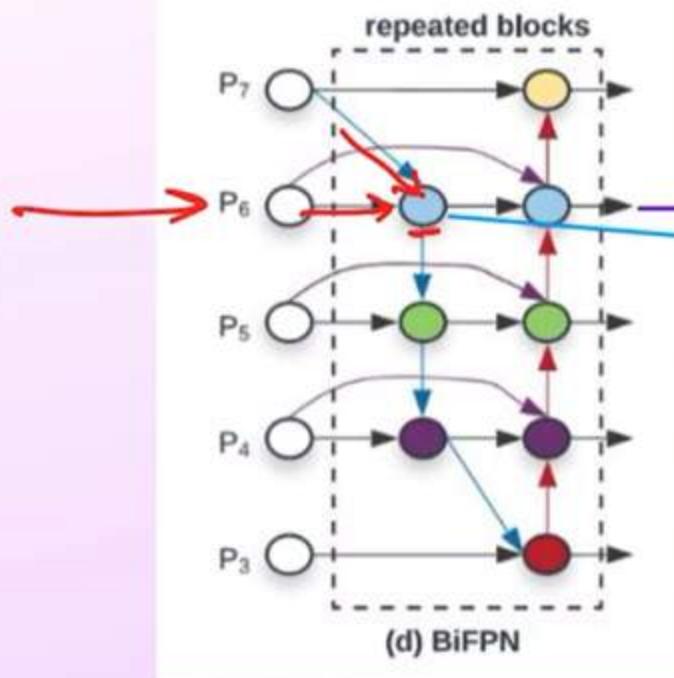
# Multi-Input Weighted Residual Connections



$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$

$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

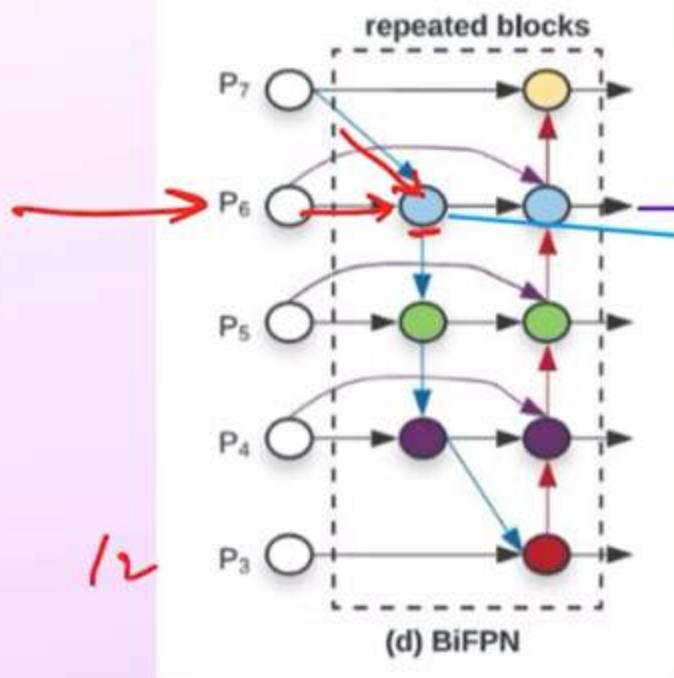
# Multi-Input Weighted Residual Connections



$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$

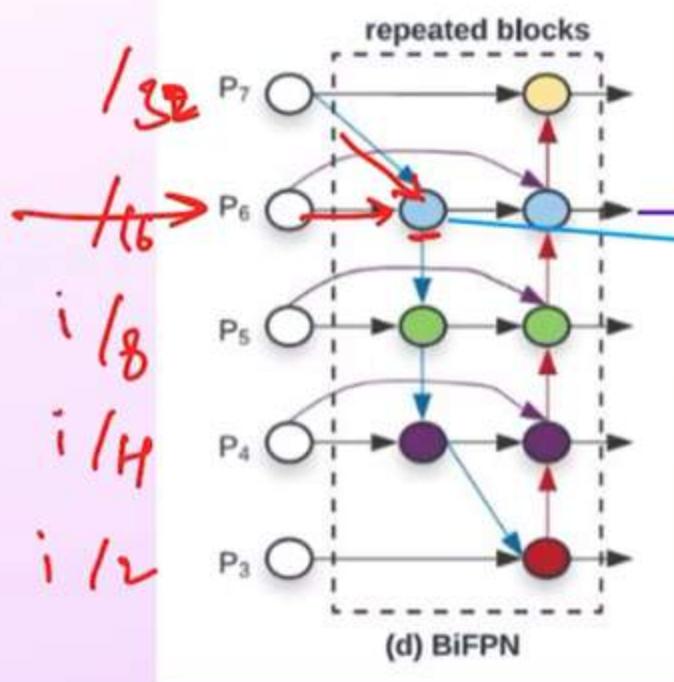
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

# Multi-Input Weighted Residual Connections



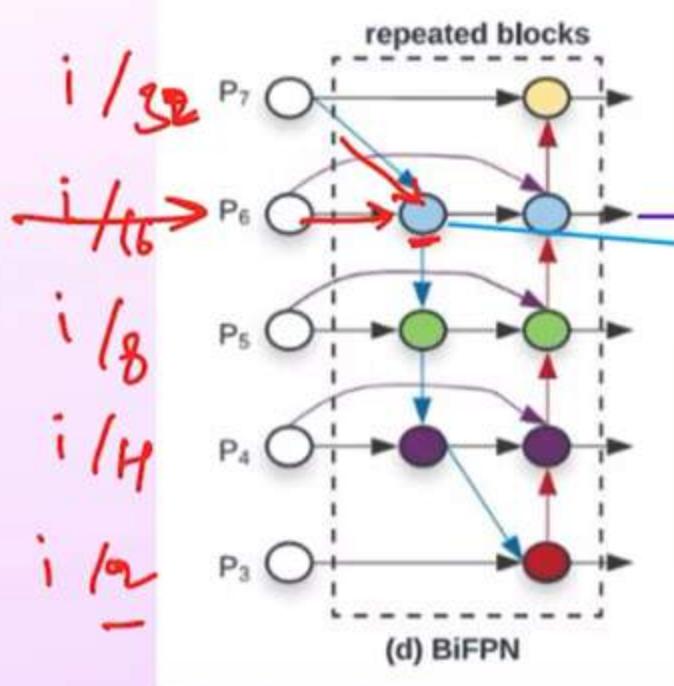
$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

# Multi-Input Weighted Residual Connections



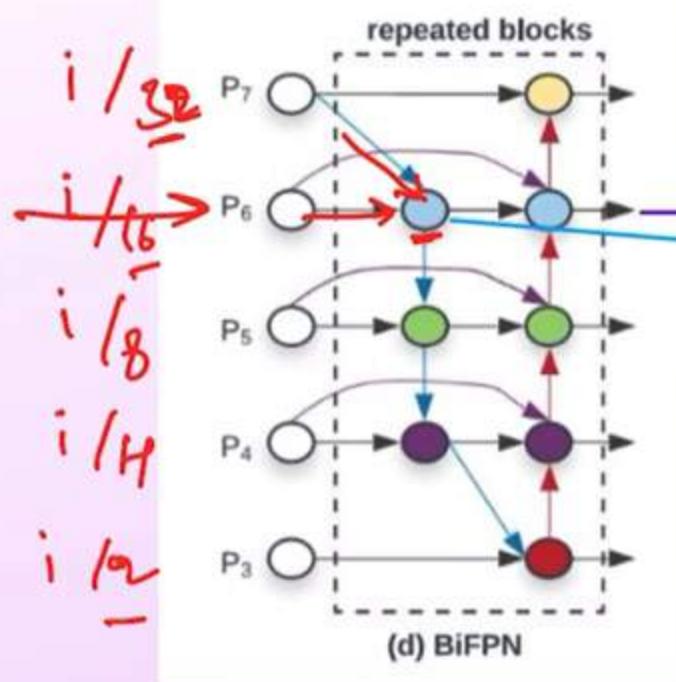
$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

# Multi-Input Weighted Residual Connections



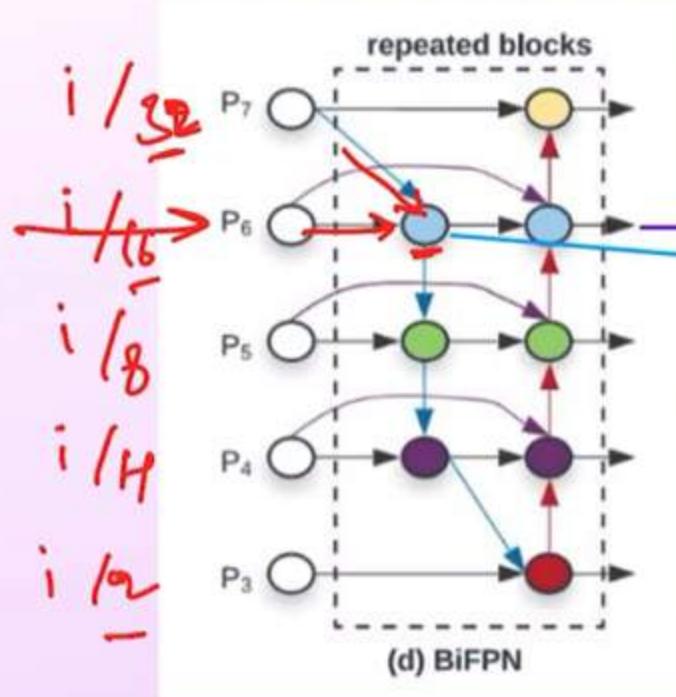
$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

# Multi-Input Weighted Residual Connections



$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

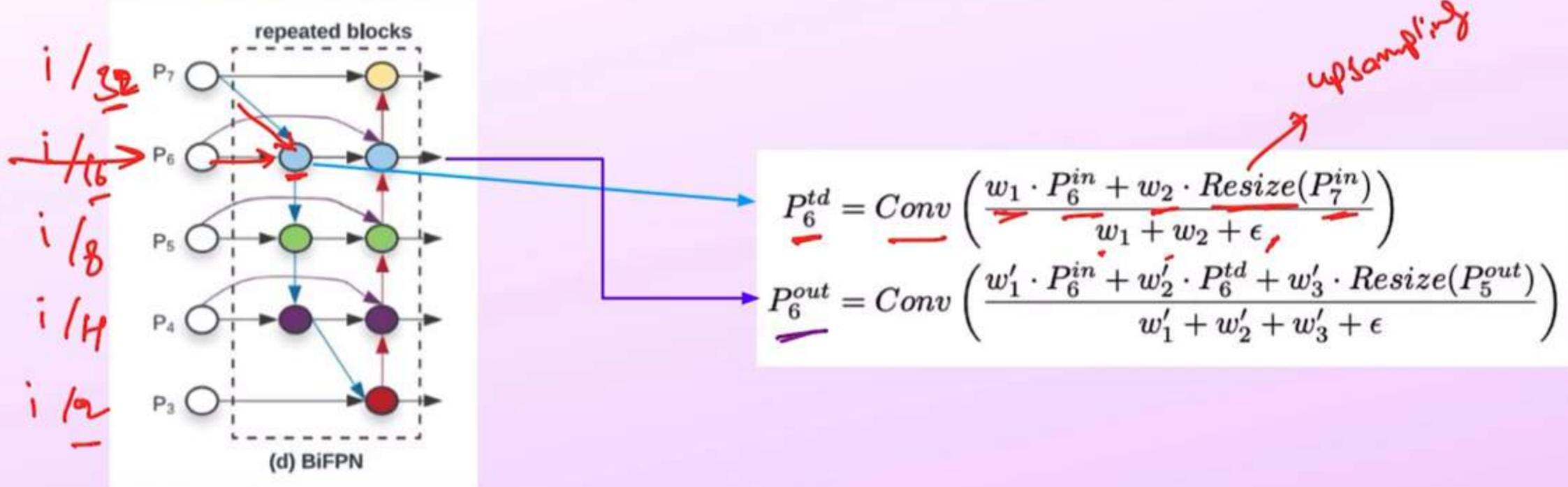
# Multi-Input Weighted Residual Connections



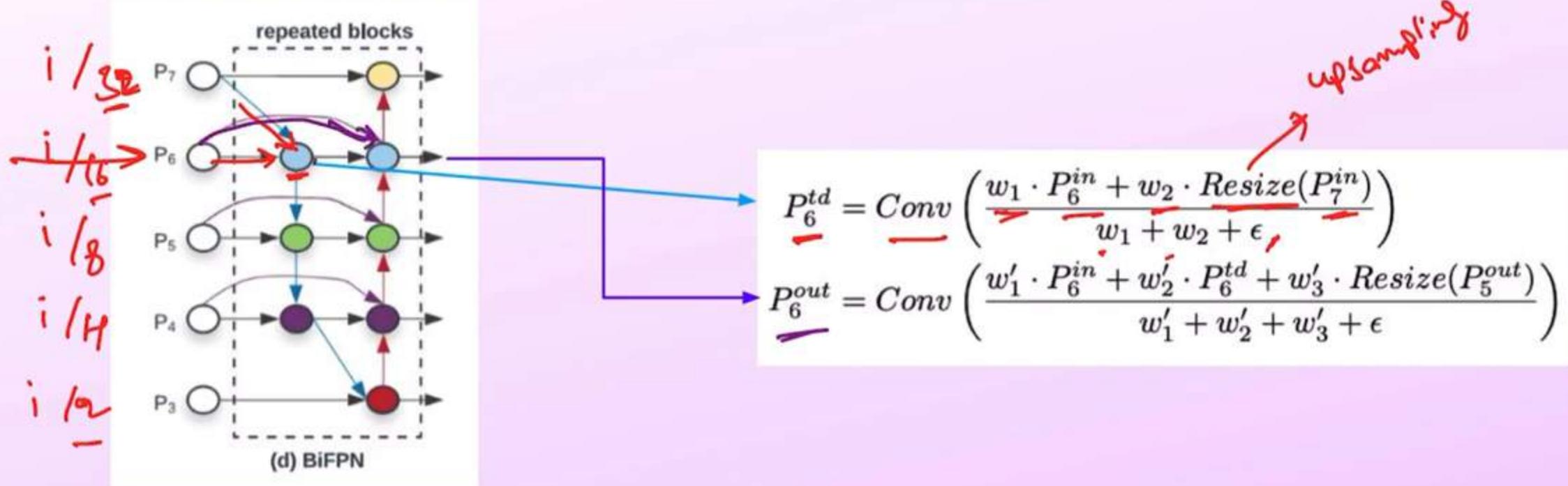
*upsampling*

$$P_6^{td} = \text{Conv} \left( \frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$
$$P_6^{out} = \text{Conv} \left( \frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot \text{Resize}(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

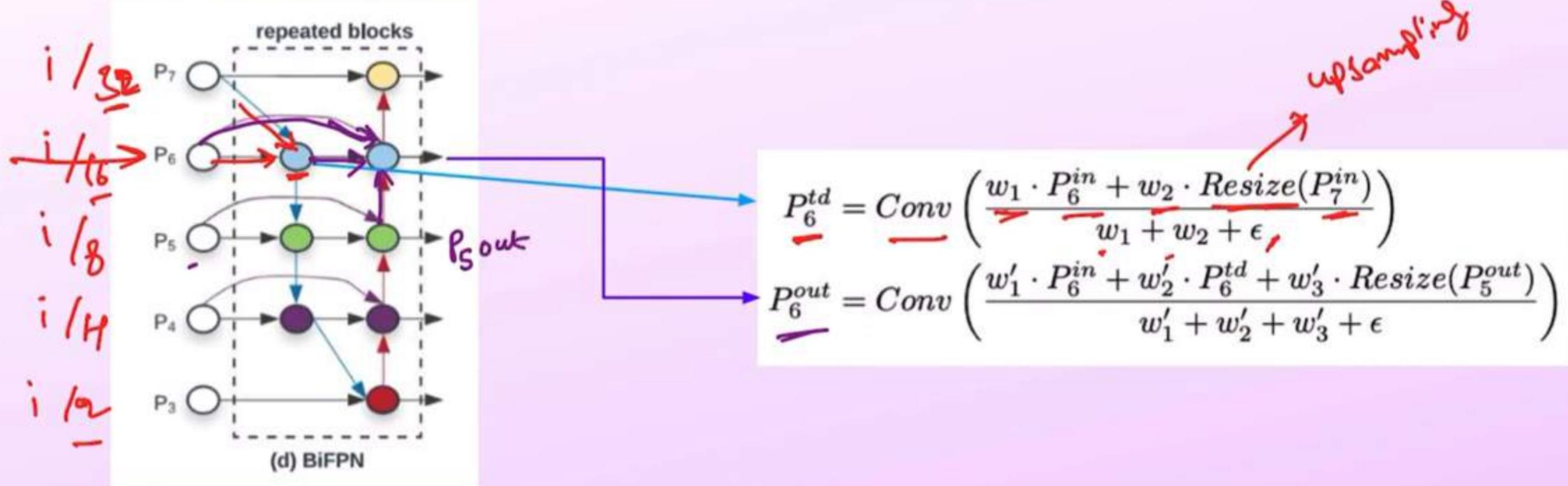
# Multi-Input Weighted Residual Connections



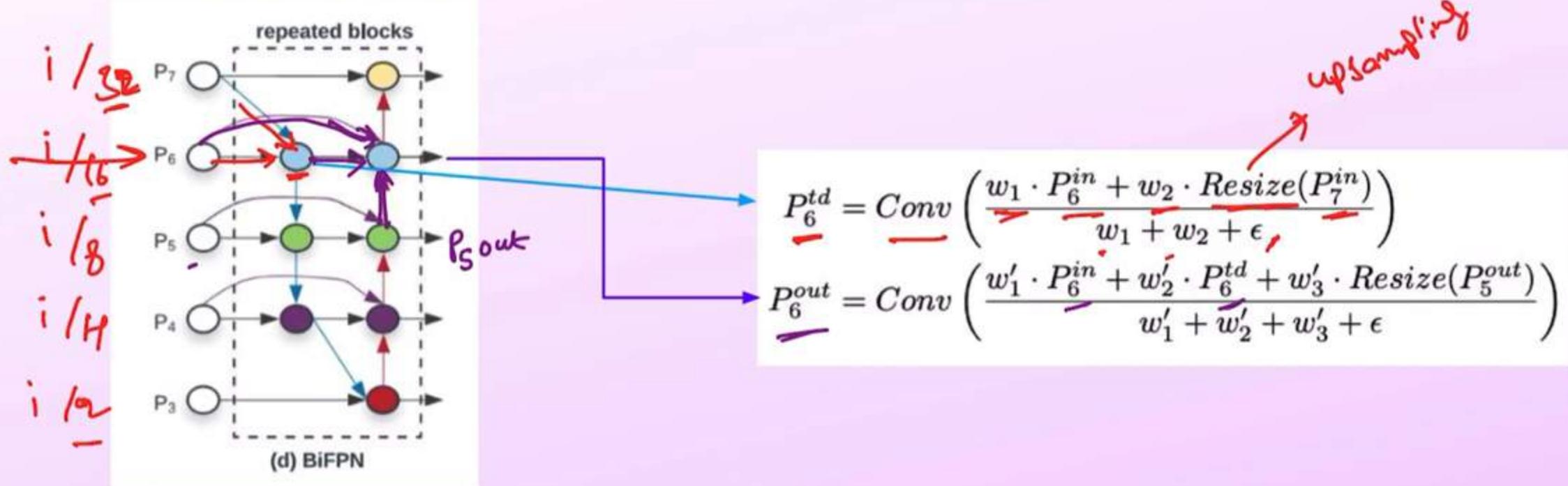
# Multi-Input Weighted Residual Connections



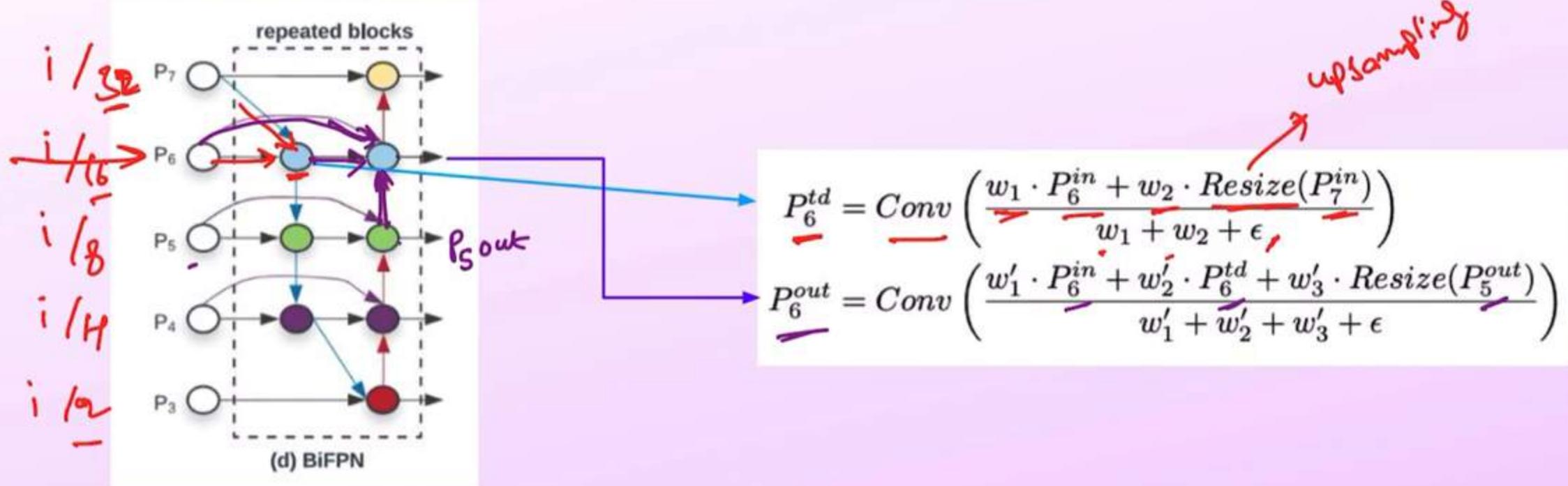
# Multi-Input Weighted Residual Connections



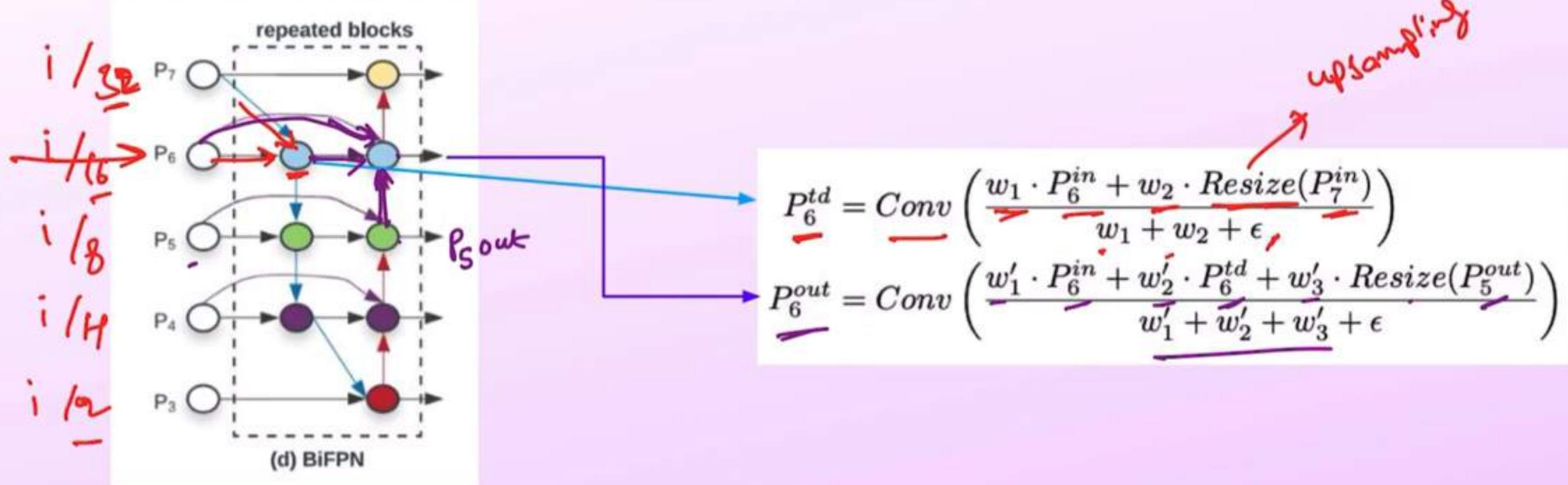
# Multi-Input Weighted Residual Connections



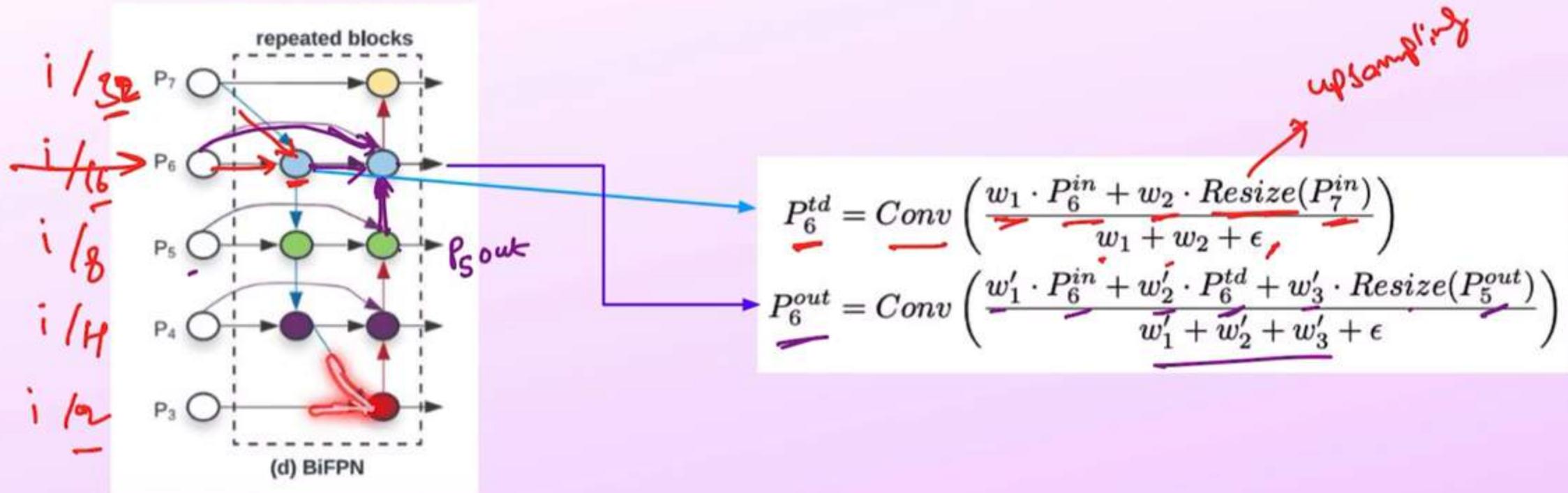
# Multi-Input Weighted Residual Connections



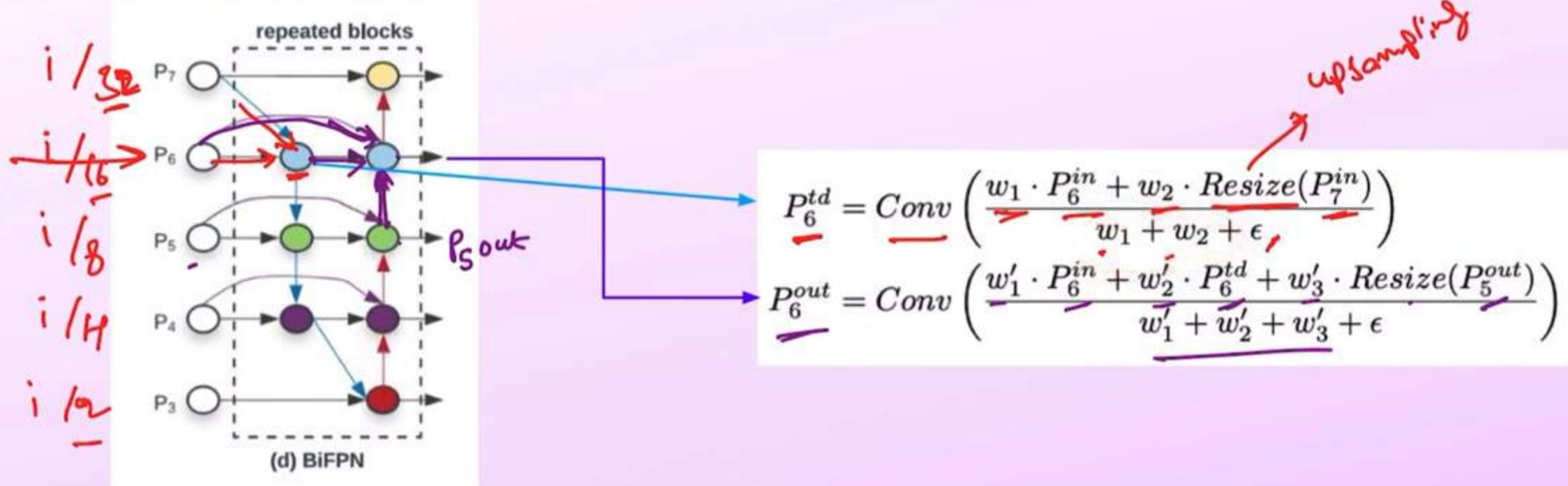
# Multi-Input Weighted Residual Connections



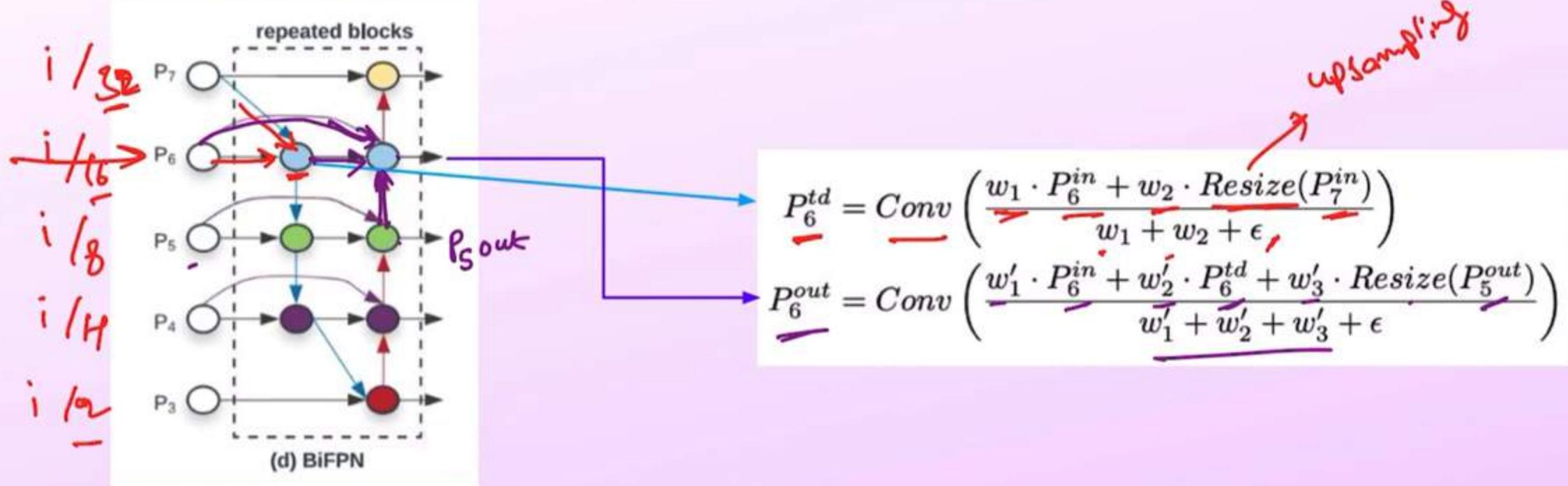
# Multi-Input Weighted Residual Connections



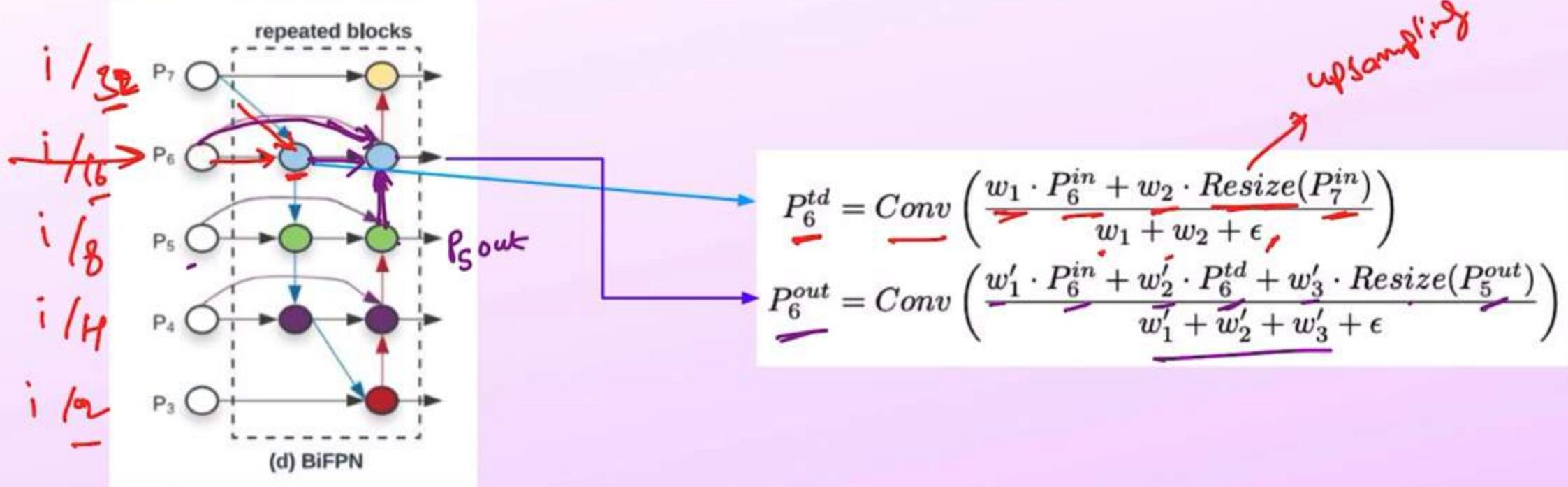
# Multi-Input Weighted Residual Connections



# Multi-Input Weighted Residual Connections

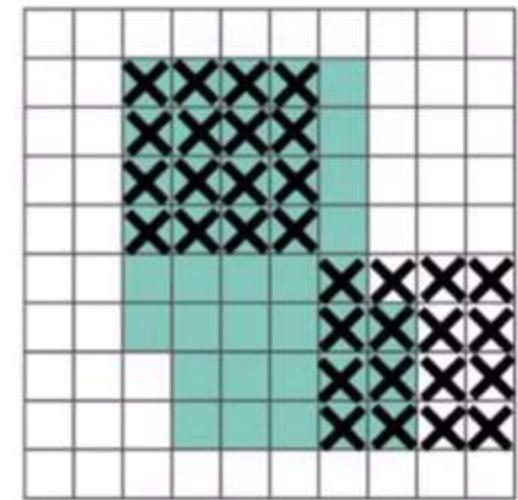
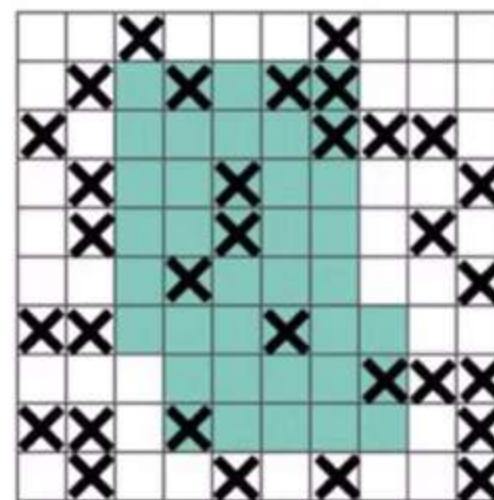


# Multi-Input Weighted Residual Connections



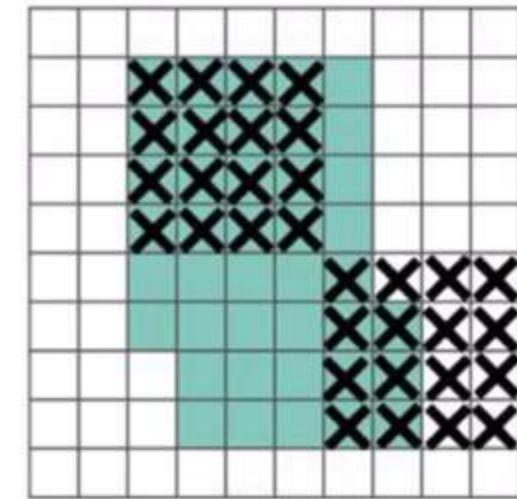
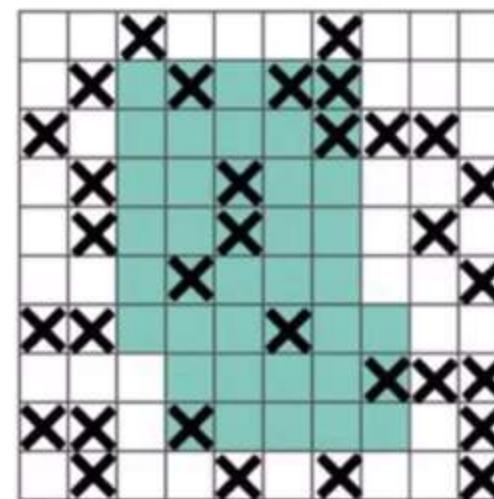
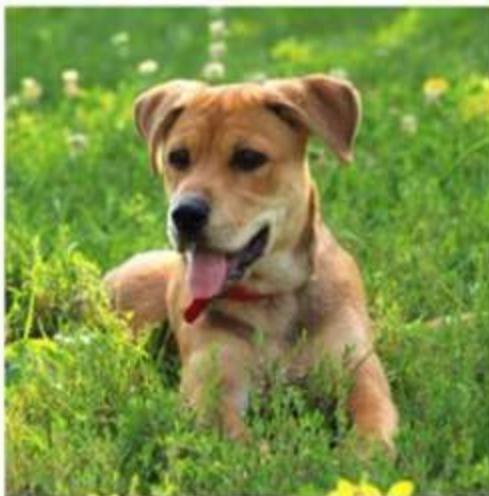
# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers



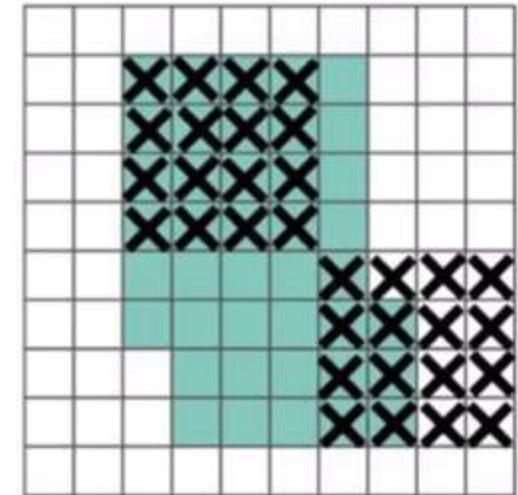
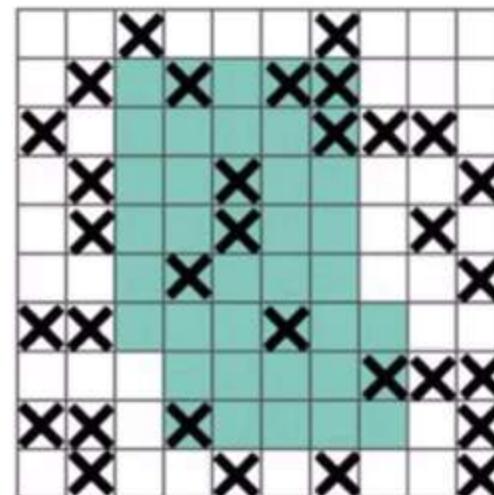
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



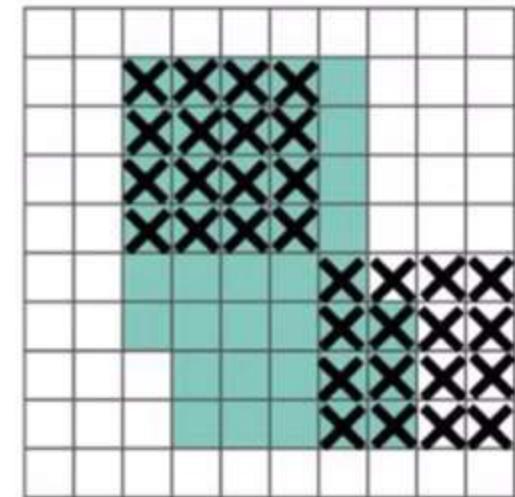
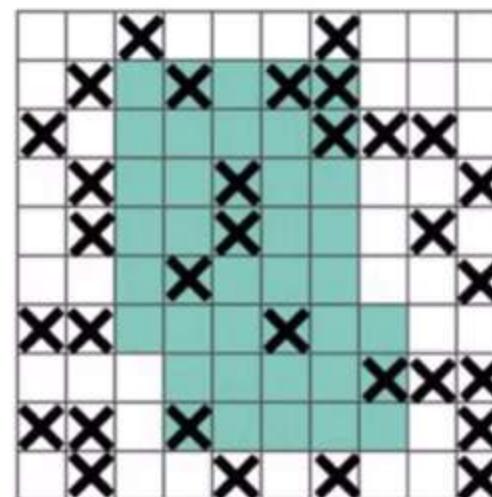
# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers



# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers



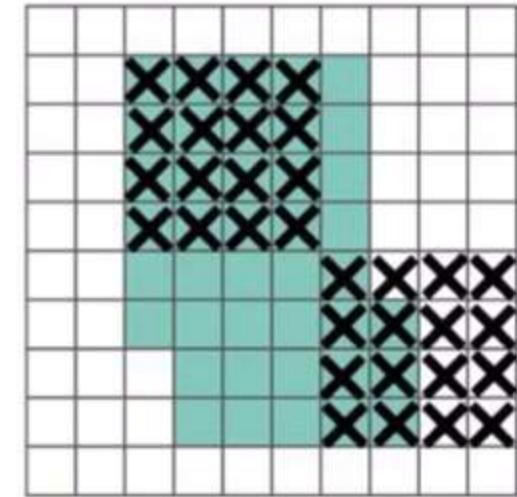
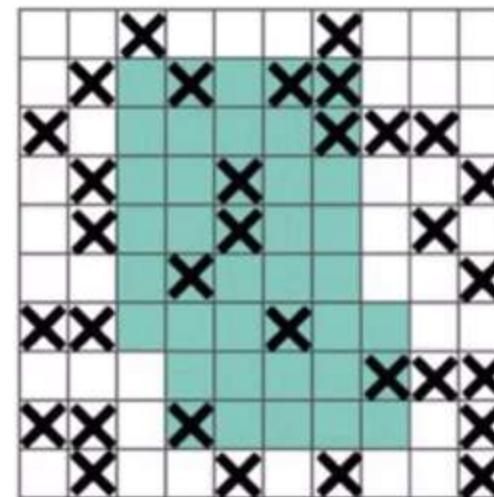
# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers

if

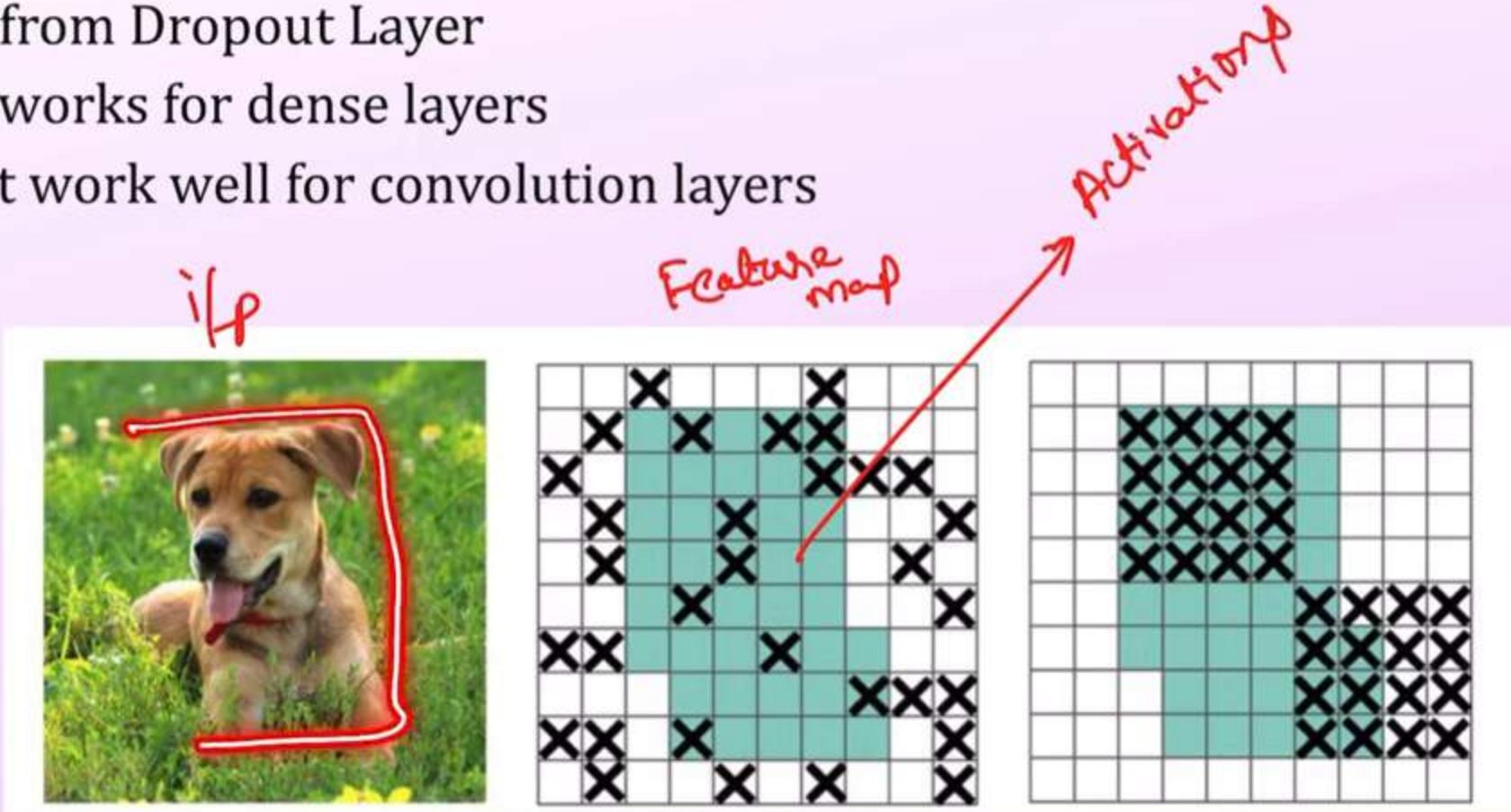


feature map



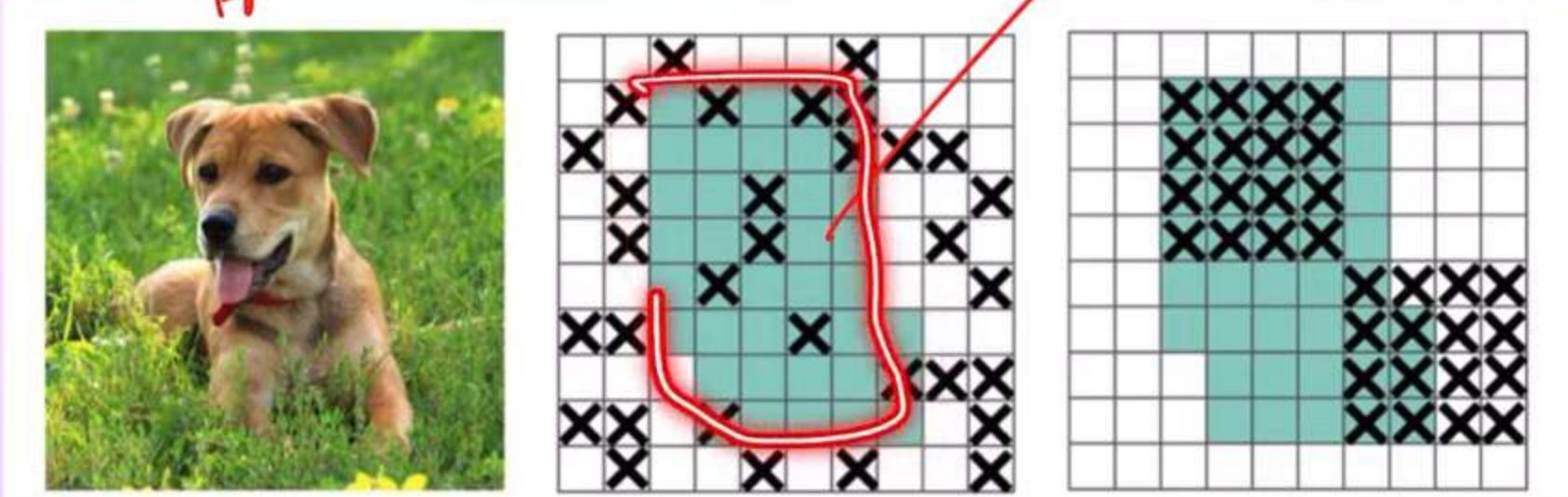
# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers



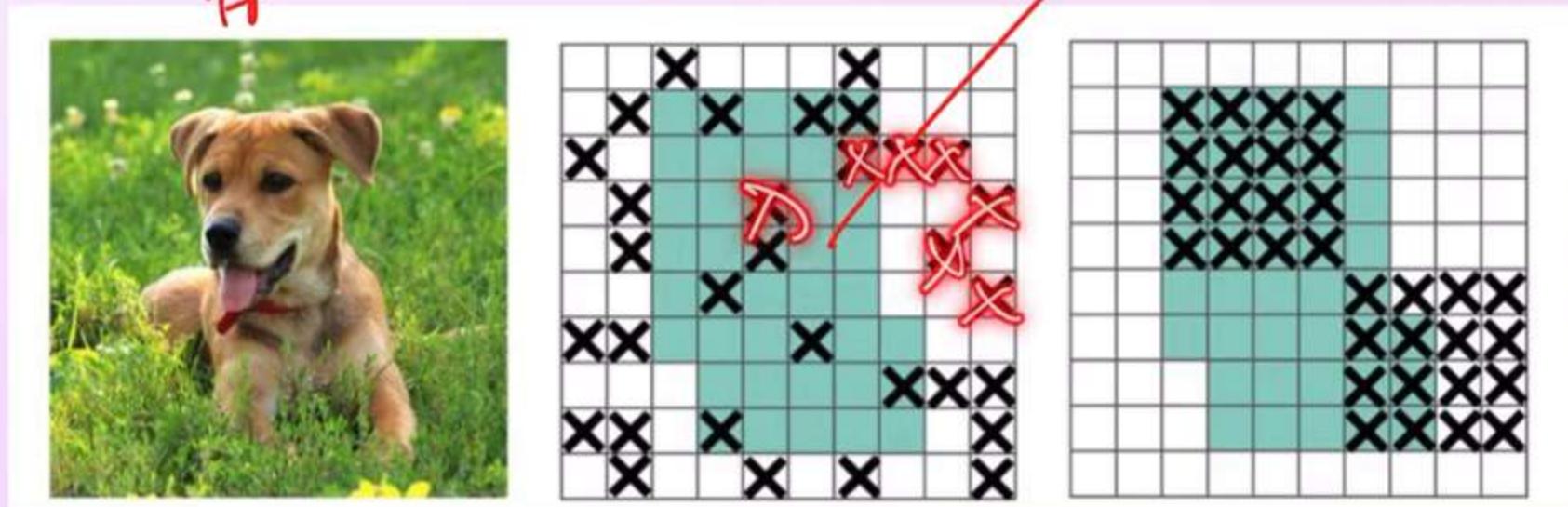
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



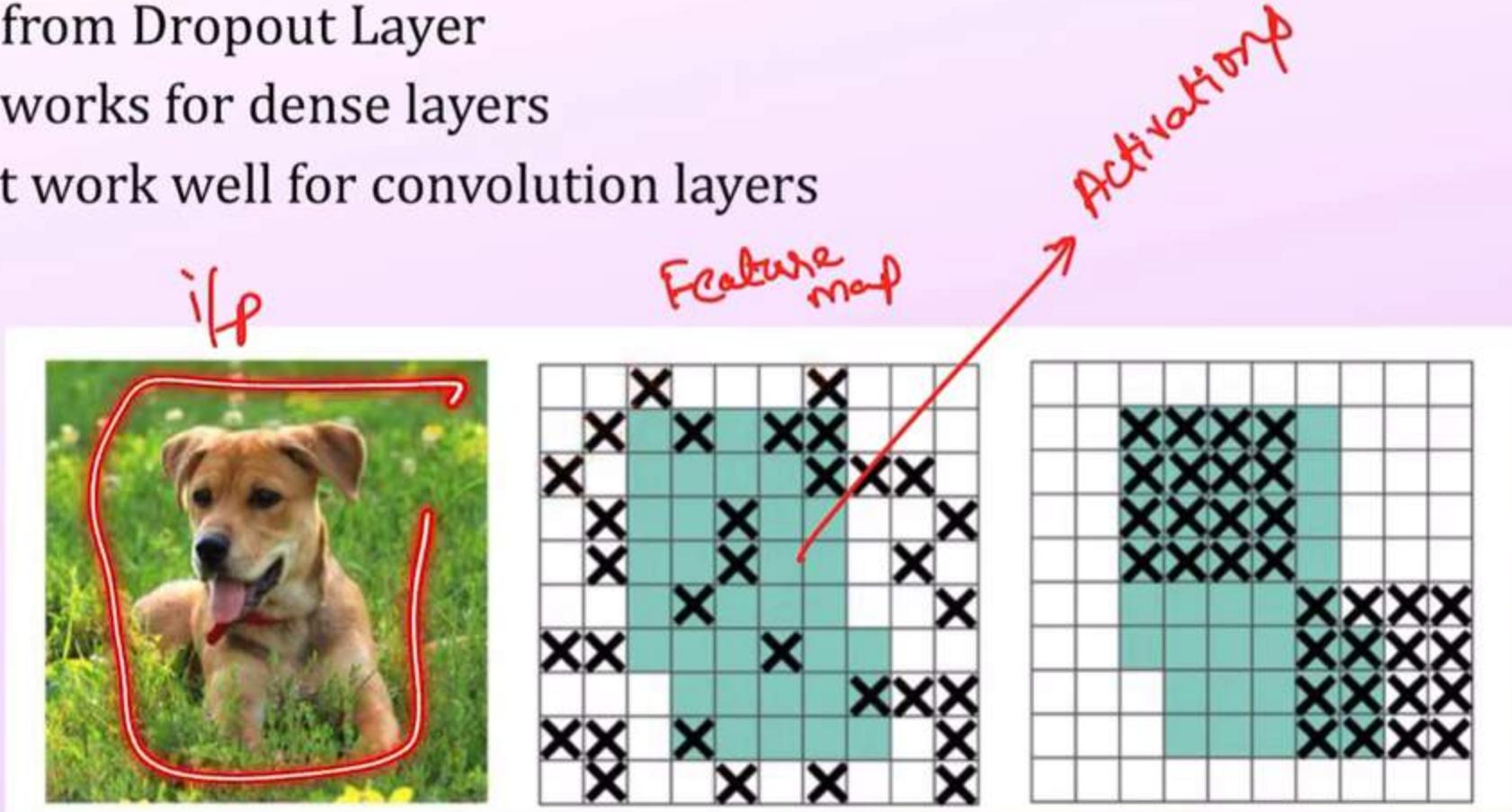
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



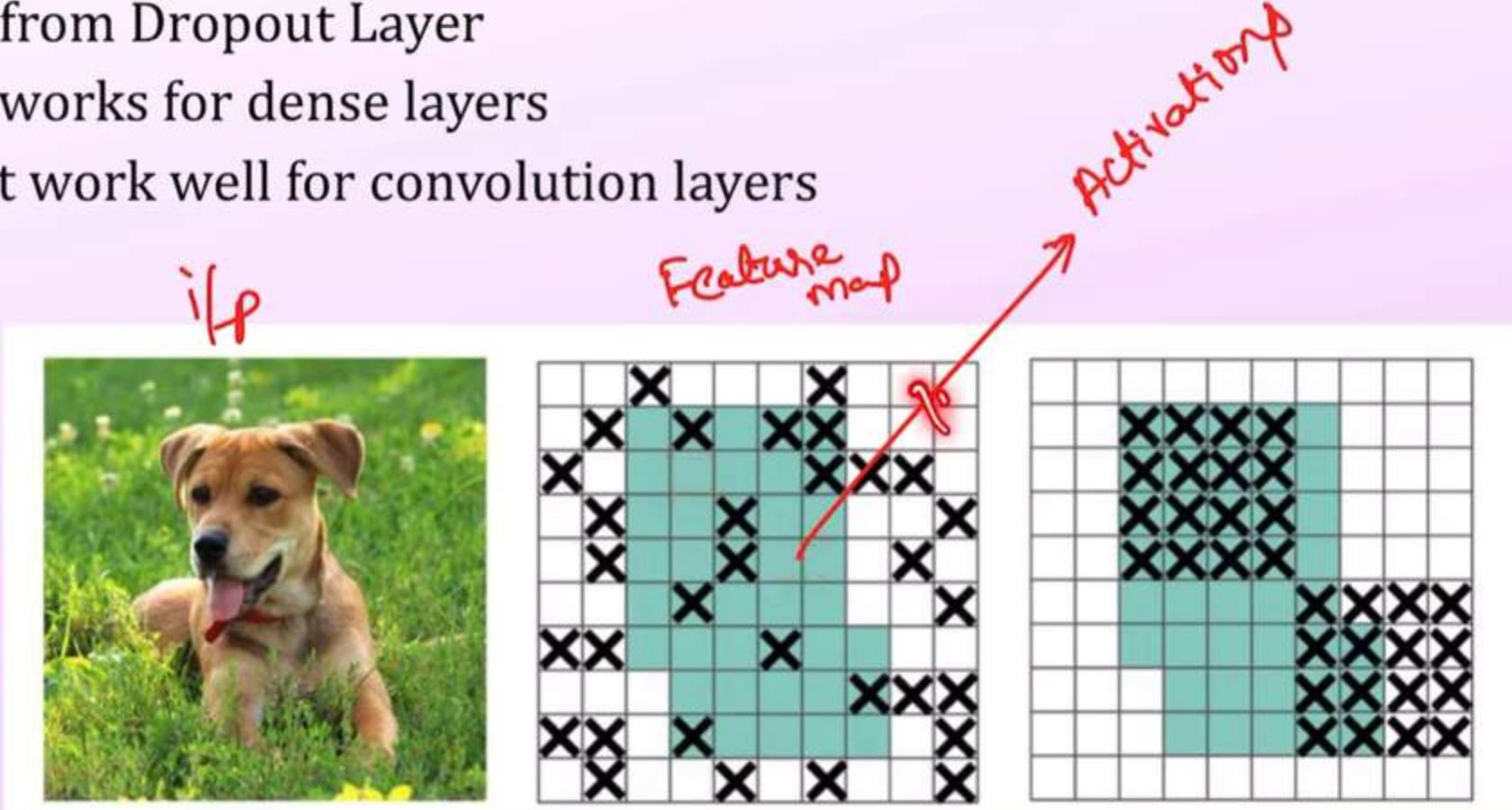
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



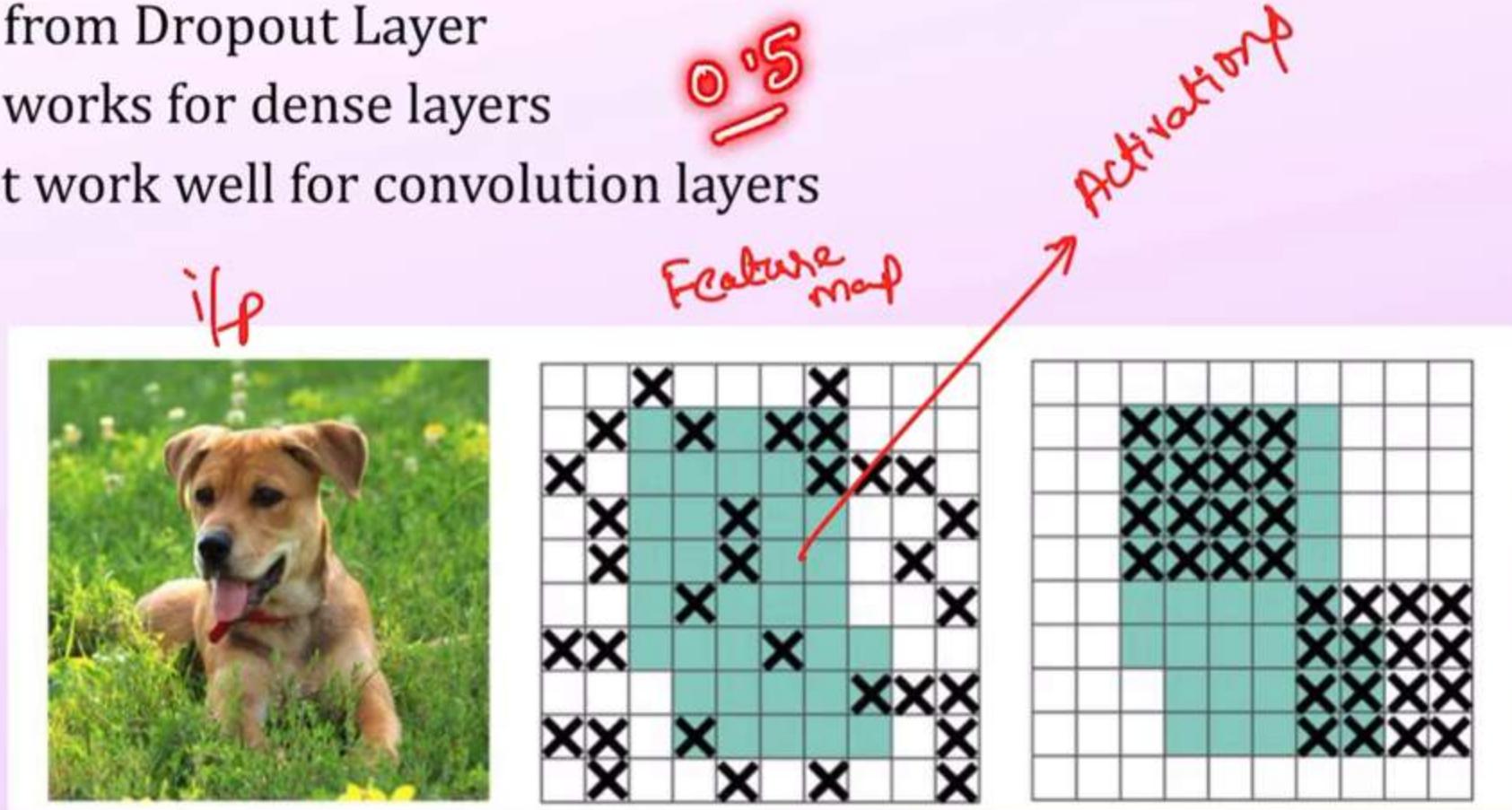
# Drop Block Regularization

- Inspired from Dropout Layer
  - Dropout works for dense layers
  - Might not work well for convolution layers



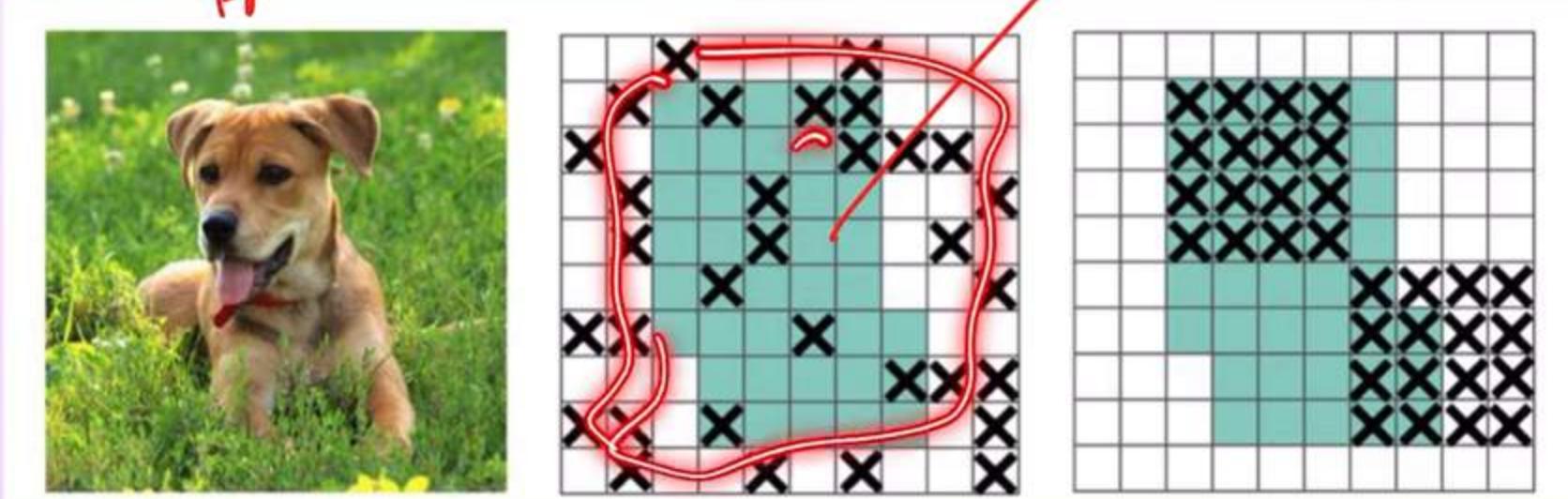
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



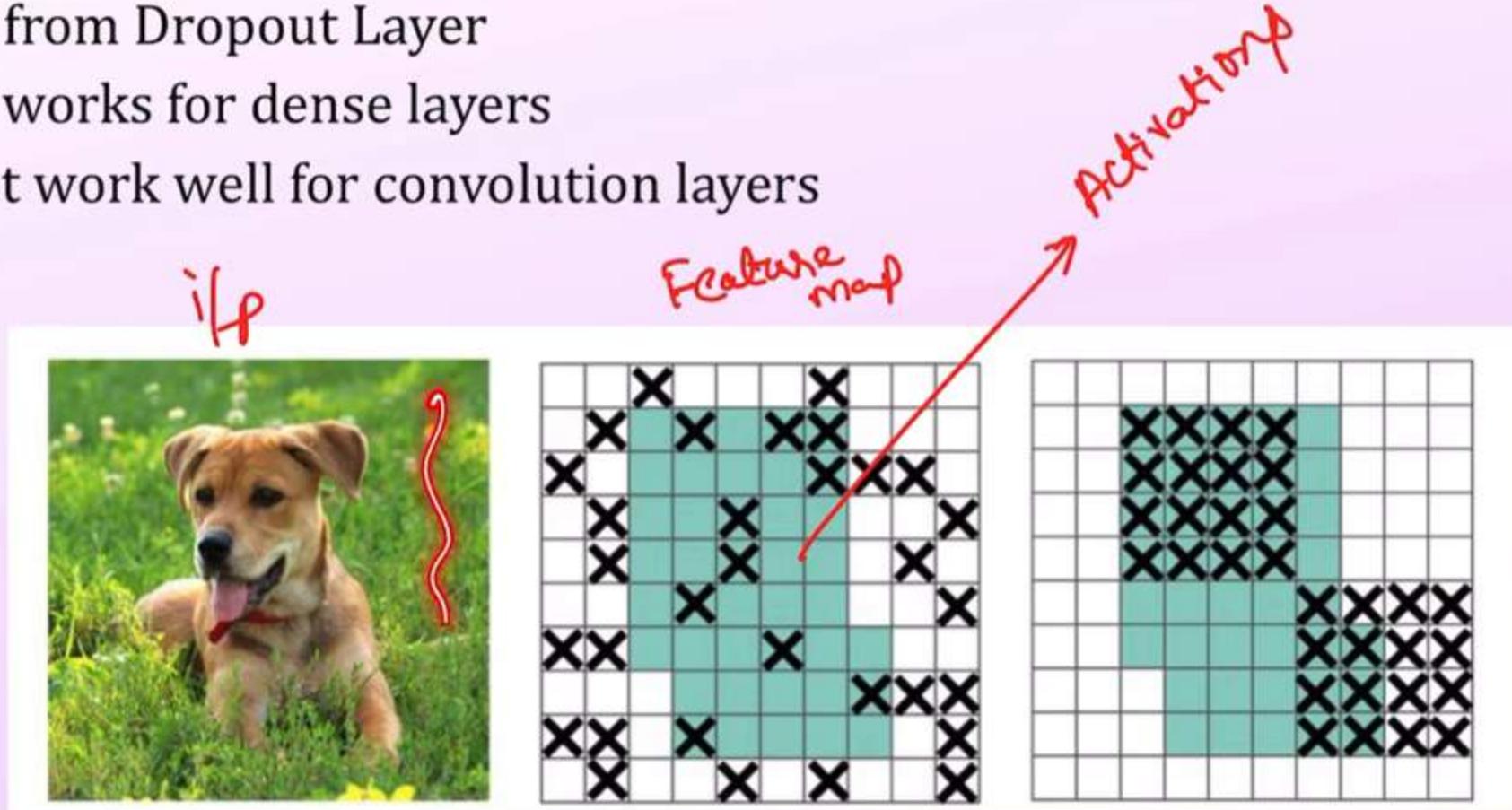
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



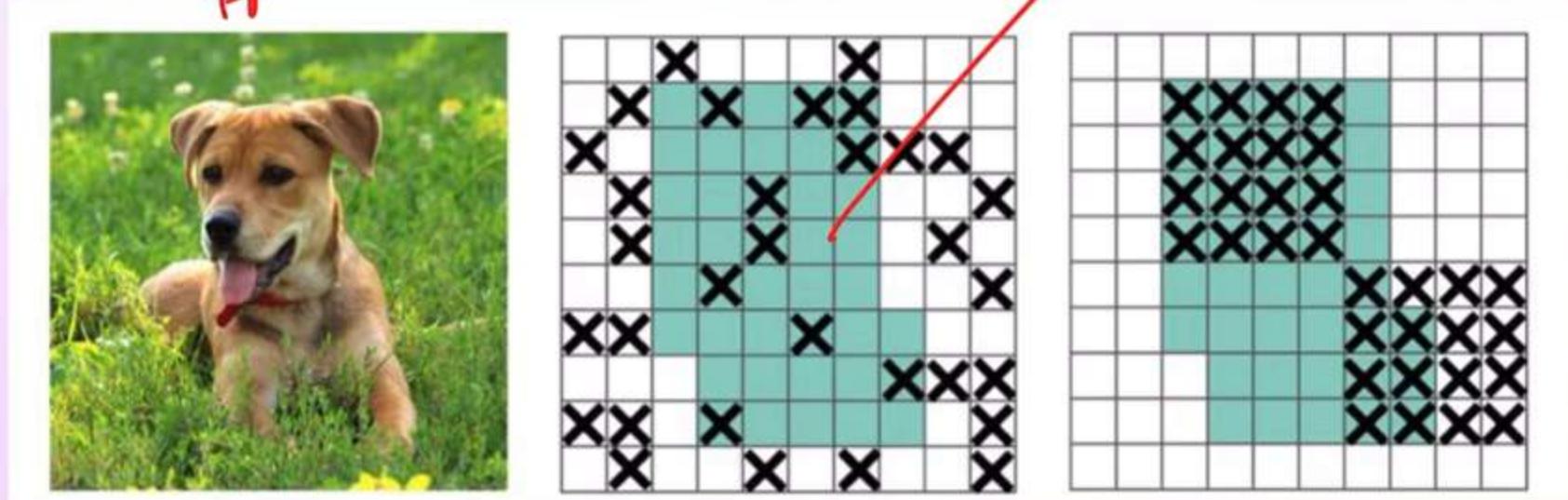
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



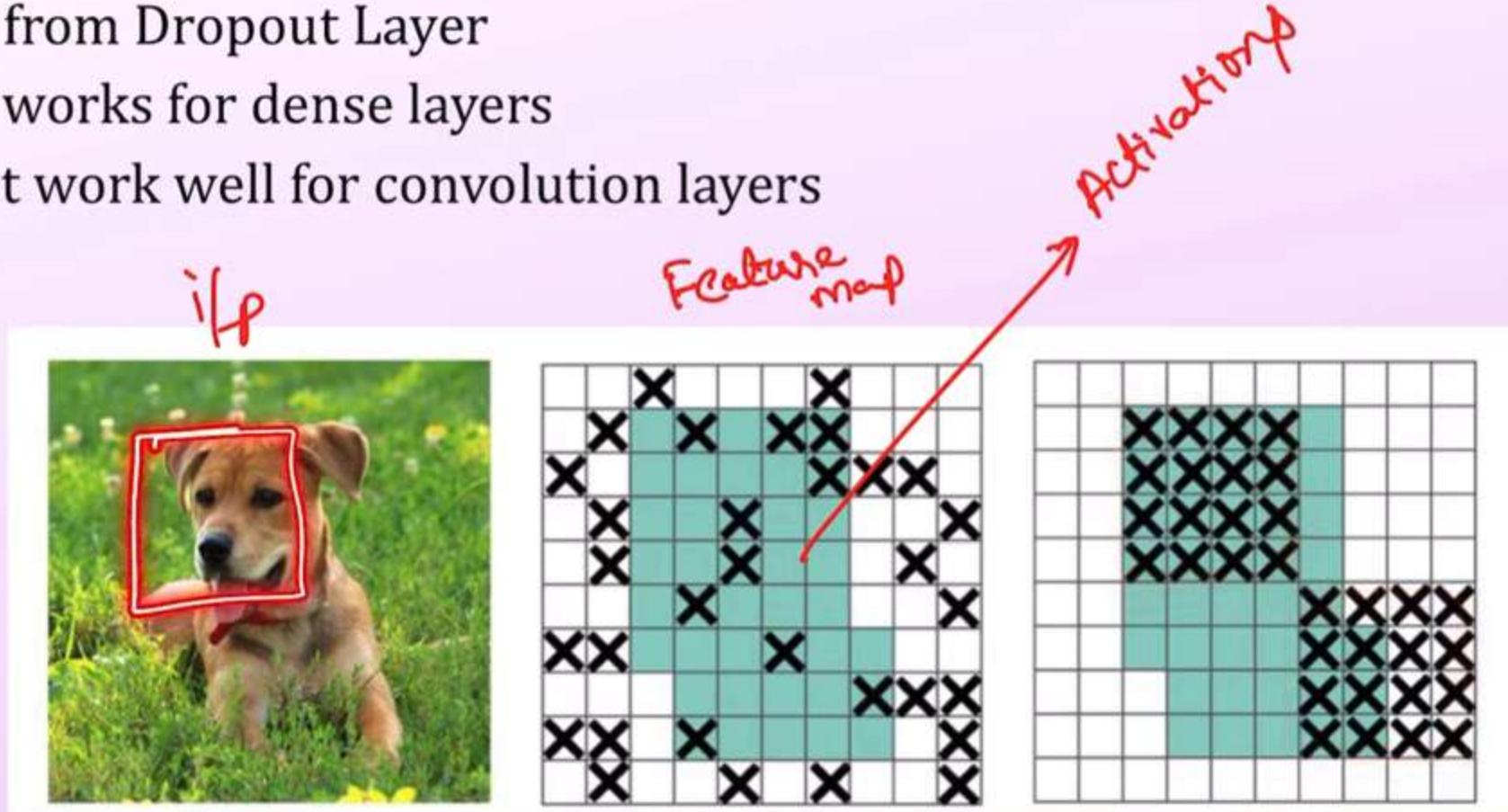
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



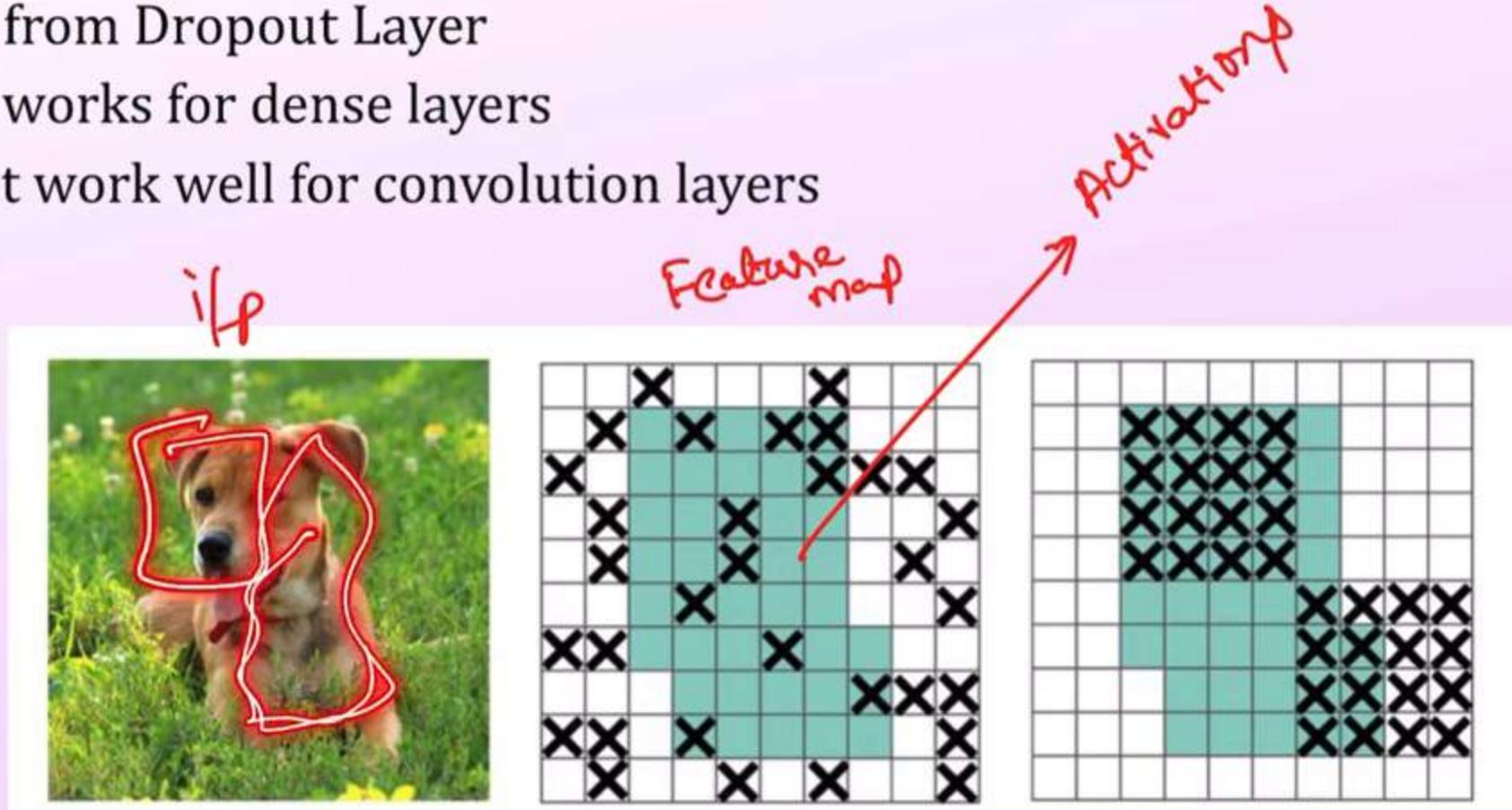
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



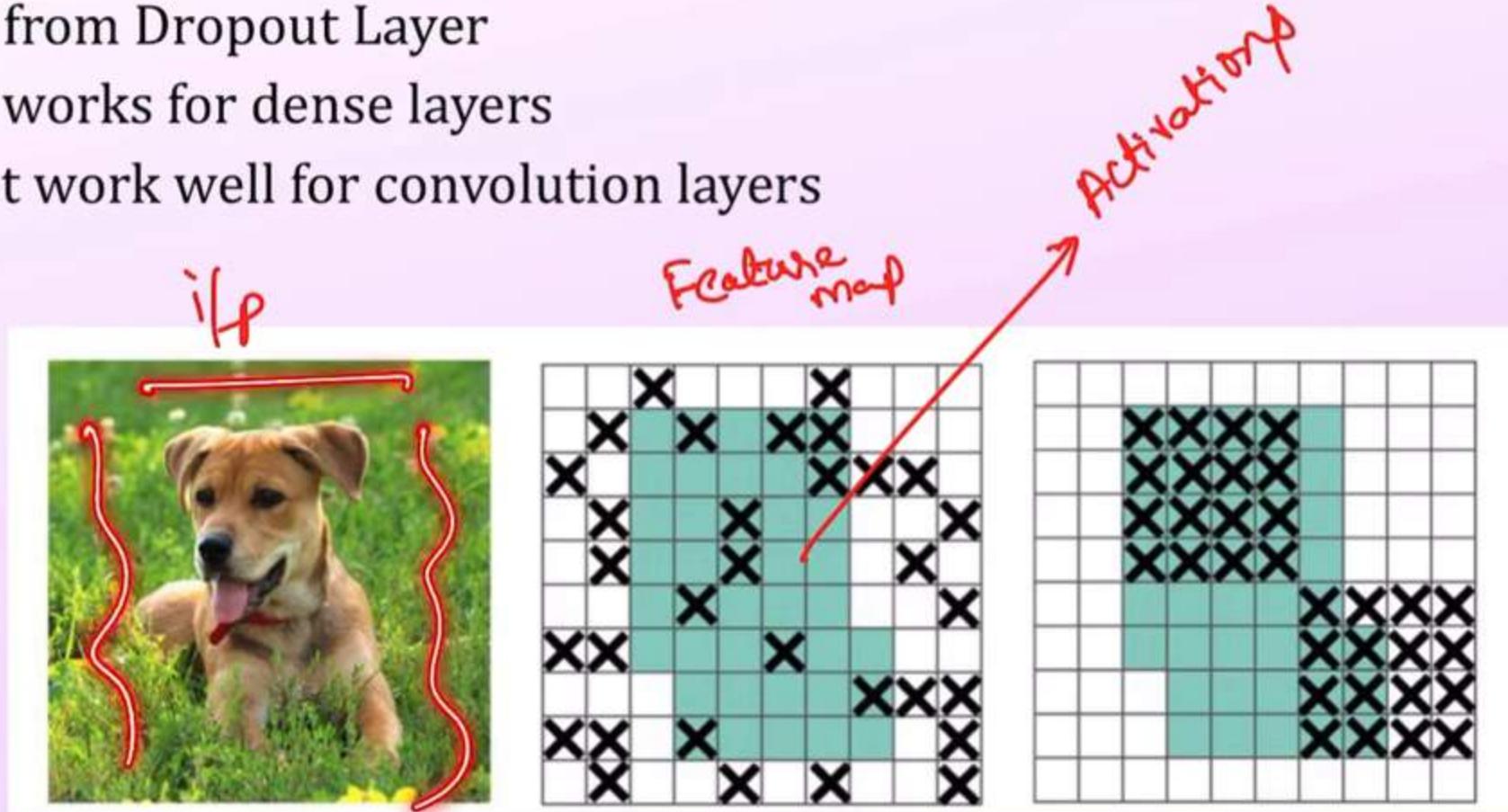
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



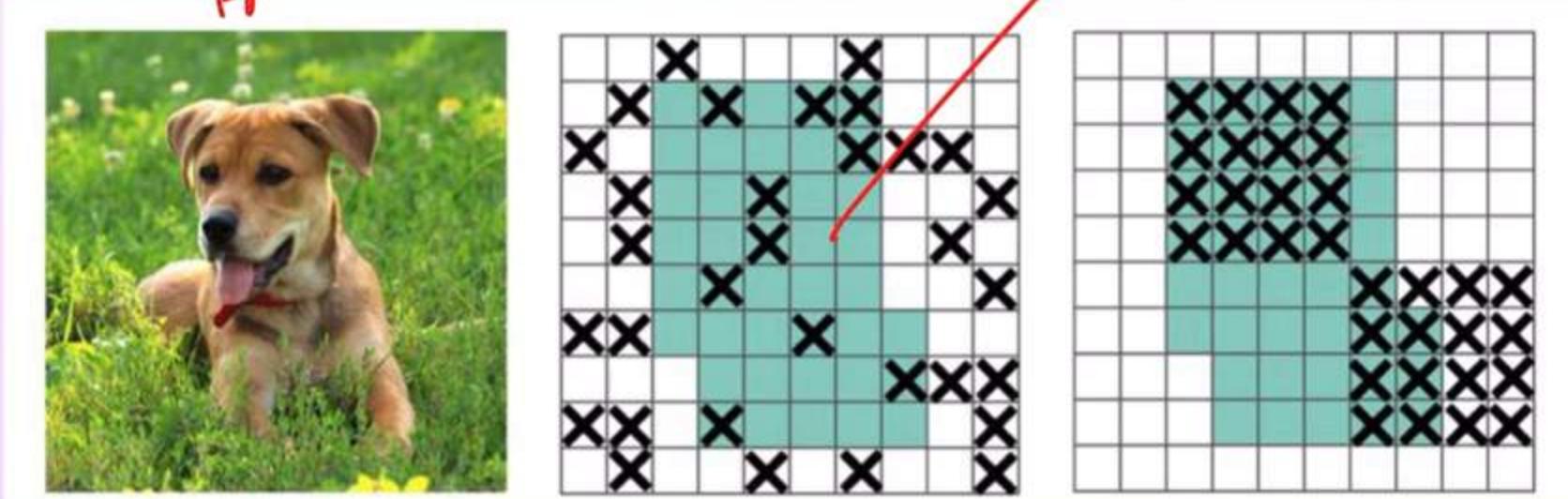
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



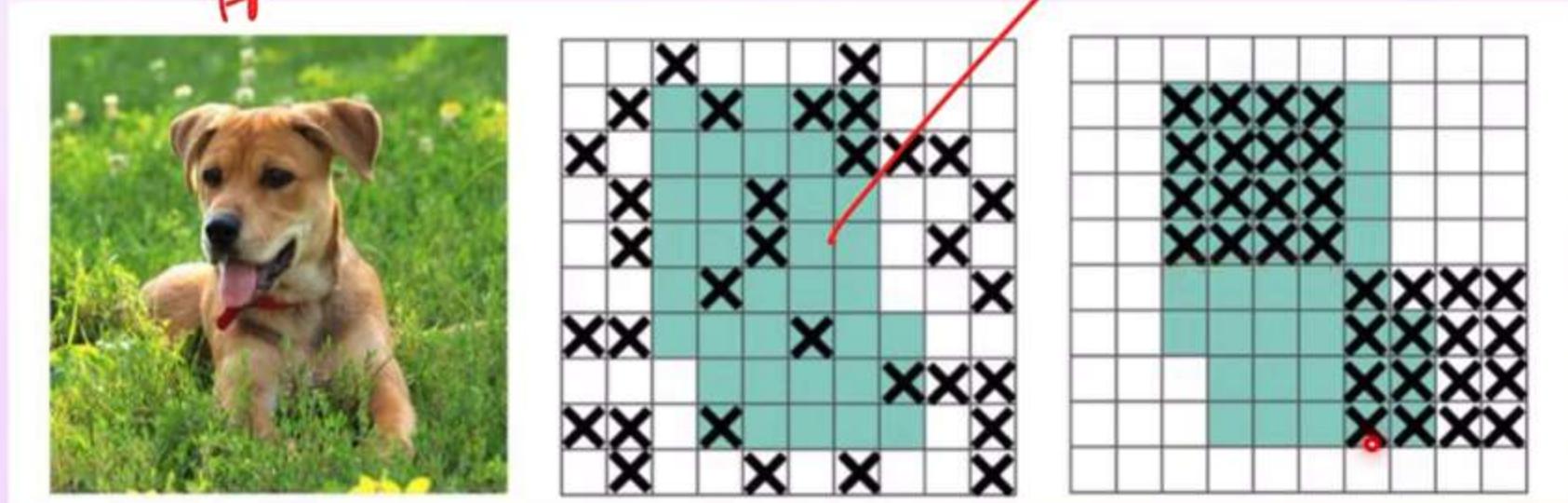
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



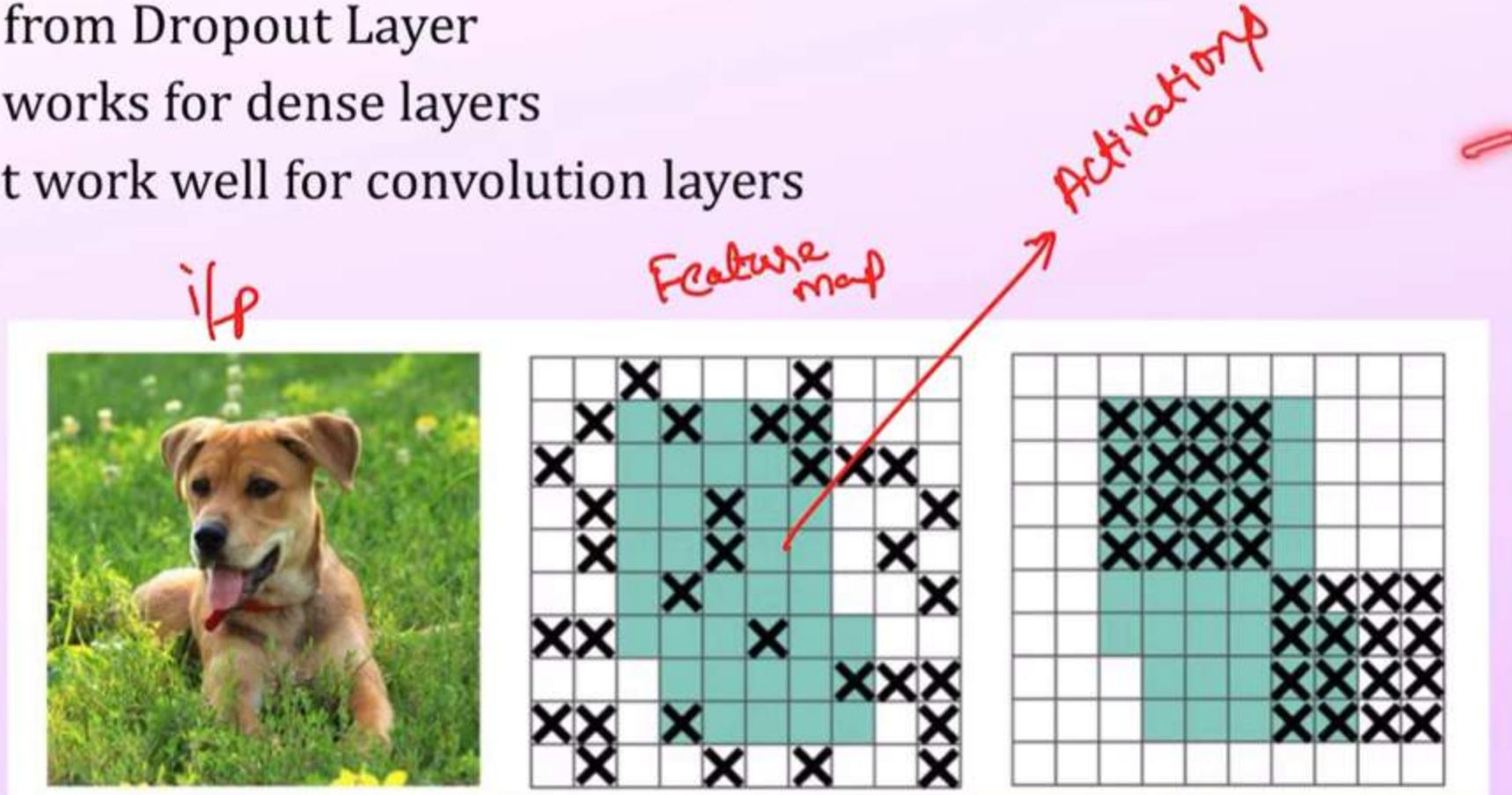
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



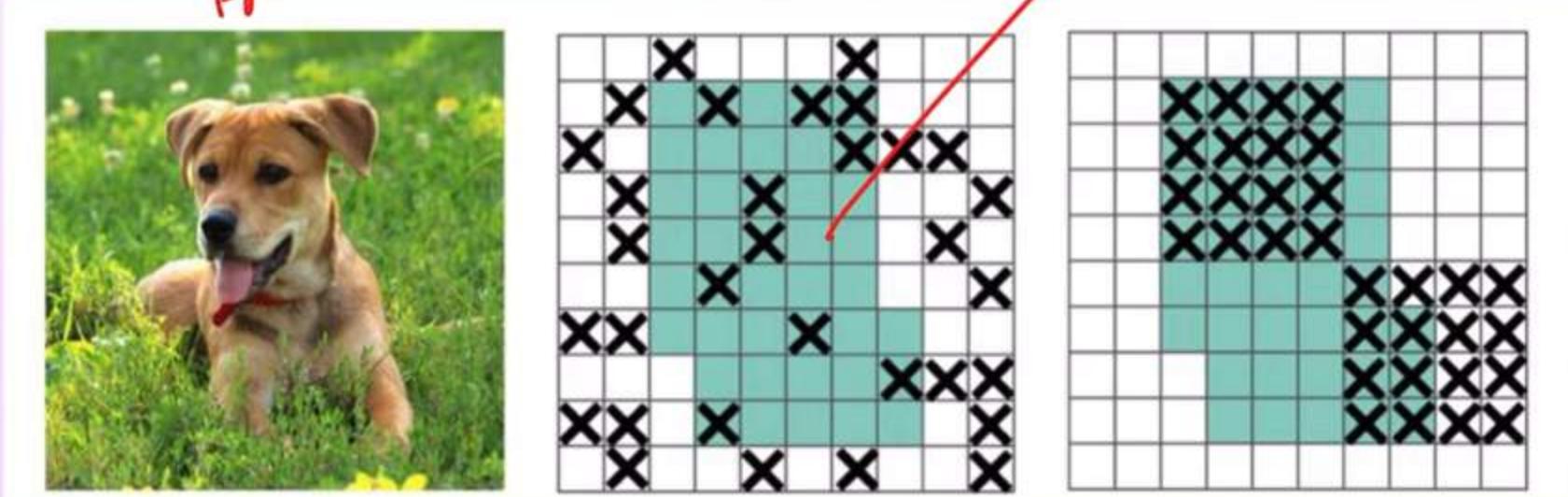
# Drop Block Regularization

- . Inspired from Dropout Layer
- . Dropout works for dense layers
- . Might not work well for convolution layers



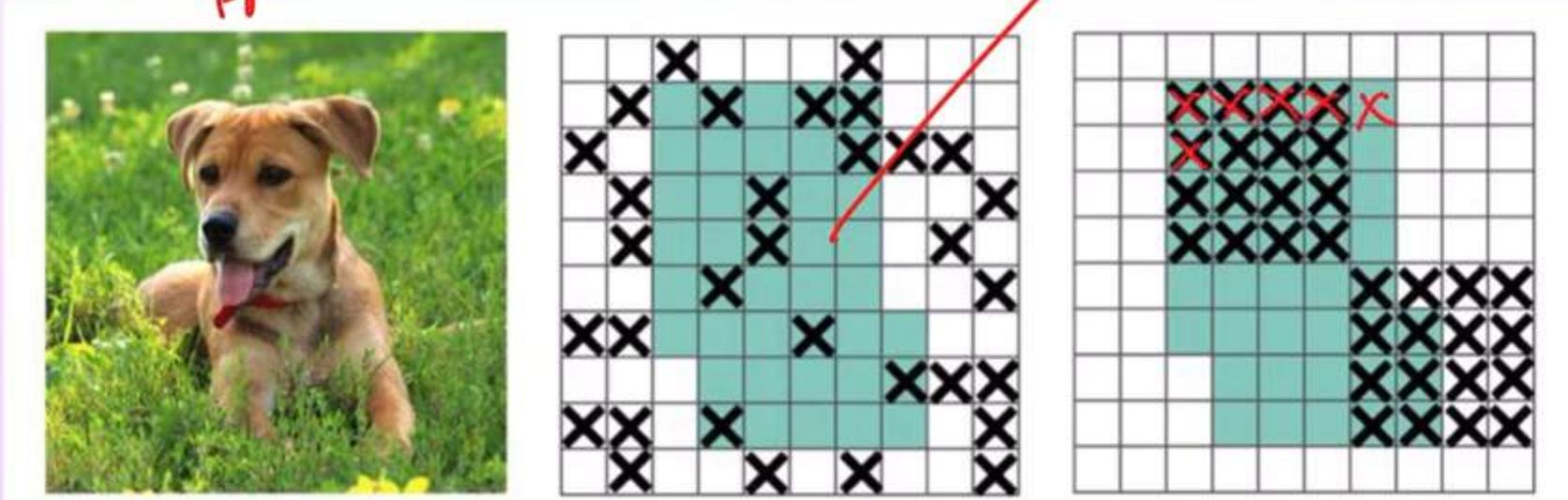
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



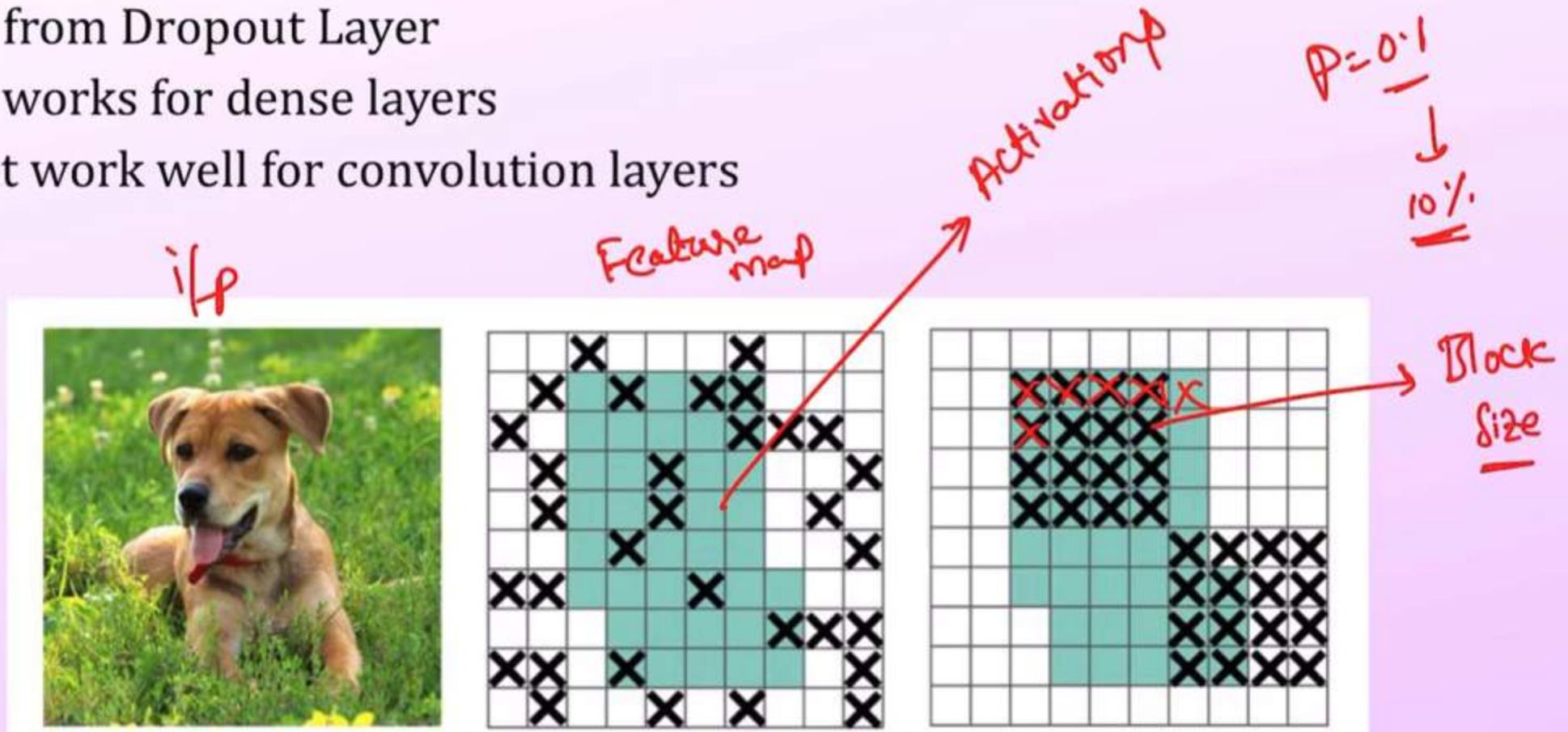
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



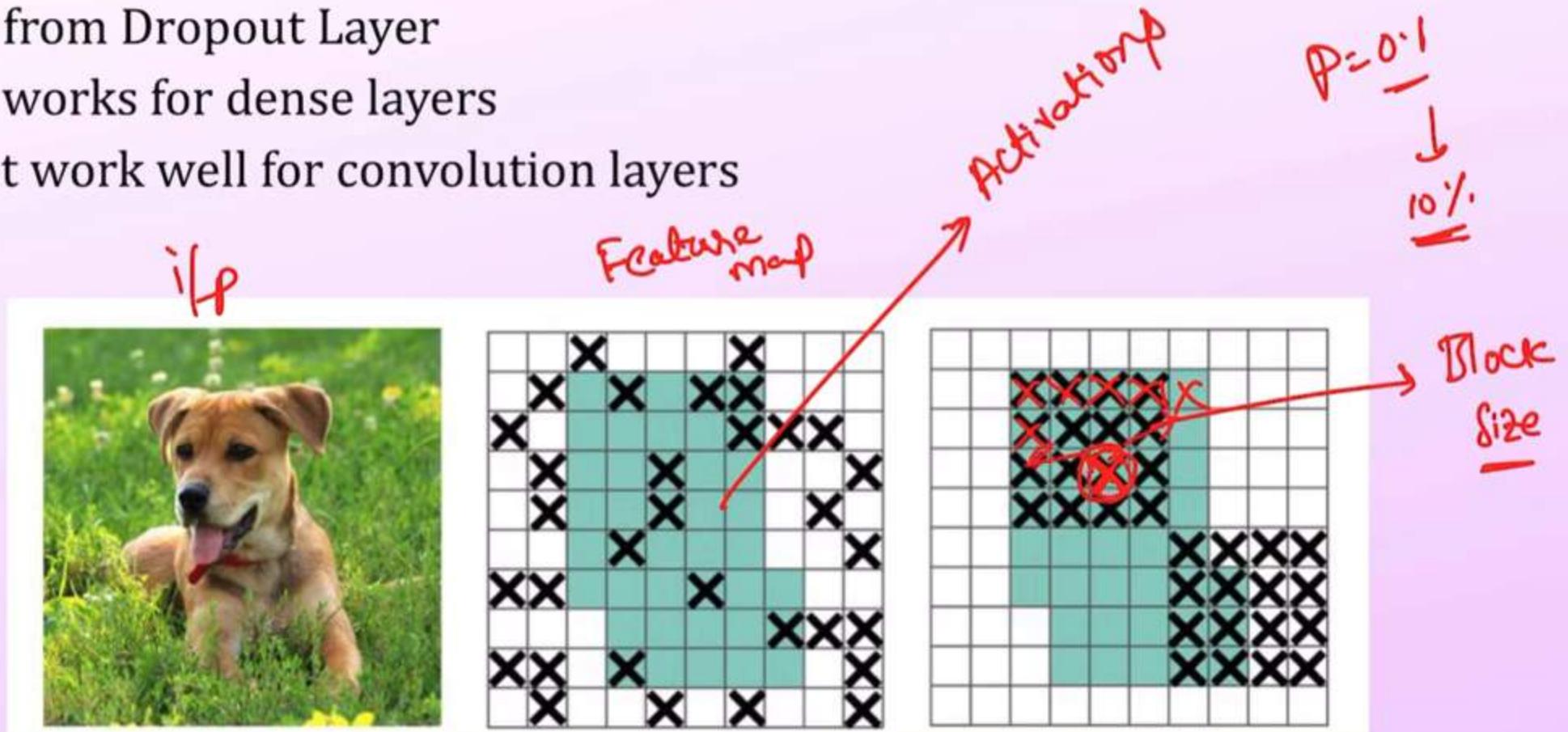
# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



# Drop Block Regularization

- Inspired from Dropout Layer
- Dropout works for dense layers
- Might not work well for convolution layers



# Drop Block Regularization

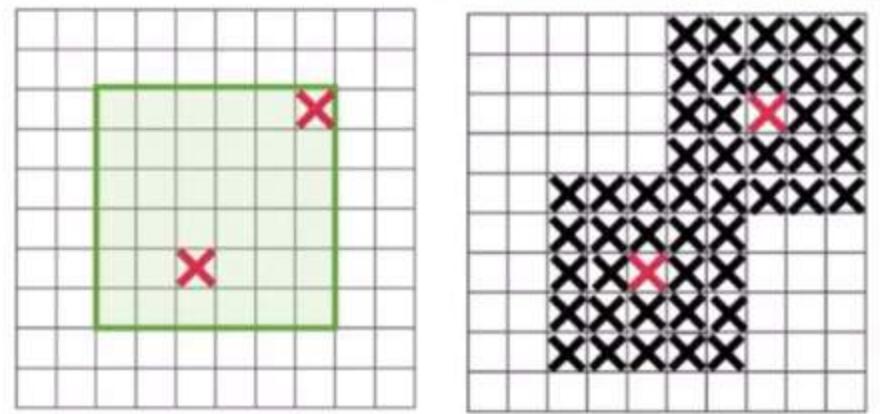
- Block\_size
- $\gamma$  - Probability of Keeping

---

**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == Inference$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim Bernoulli(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
- 



# Drop Block Regularization

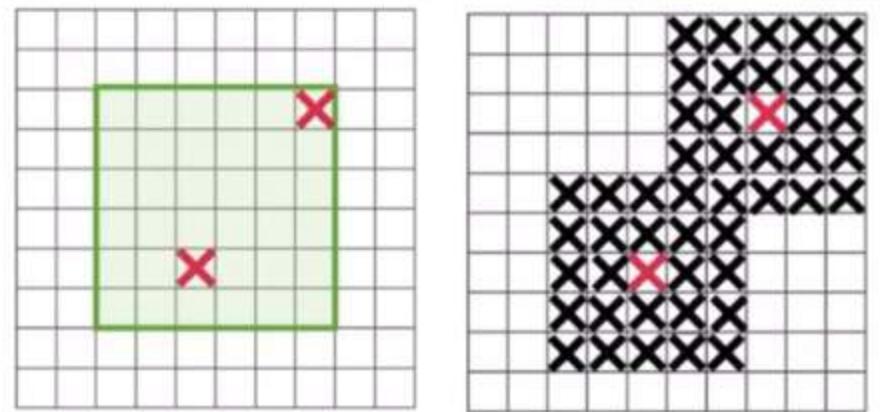
- `Block_size`
- $\gamma$  - Probability  
of Keeping

---

**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
- 



# Drop Block Regularization

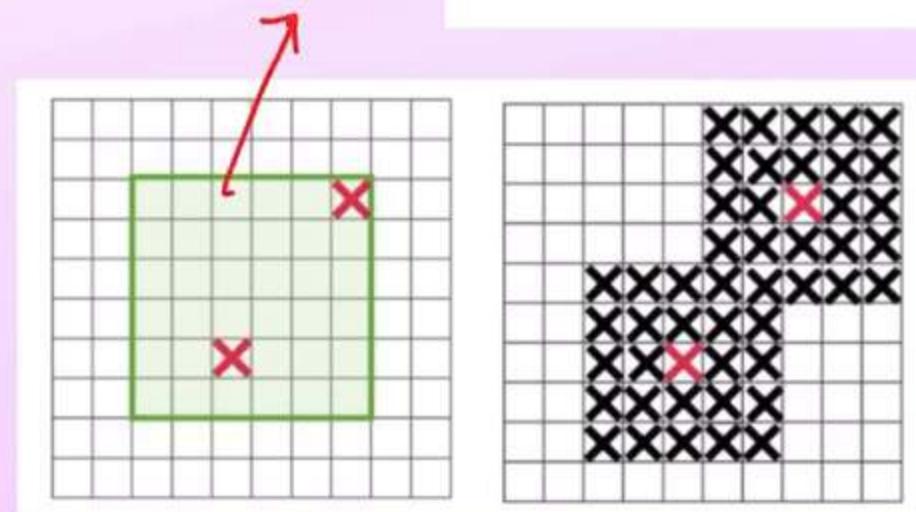
- Block\_size
- $\gamma$  - Probability of Keeping

---

**Algorithm 1** DropBlock

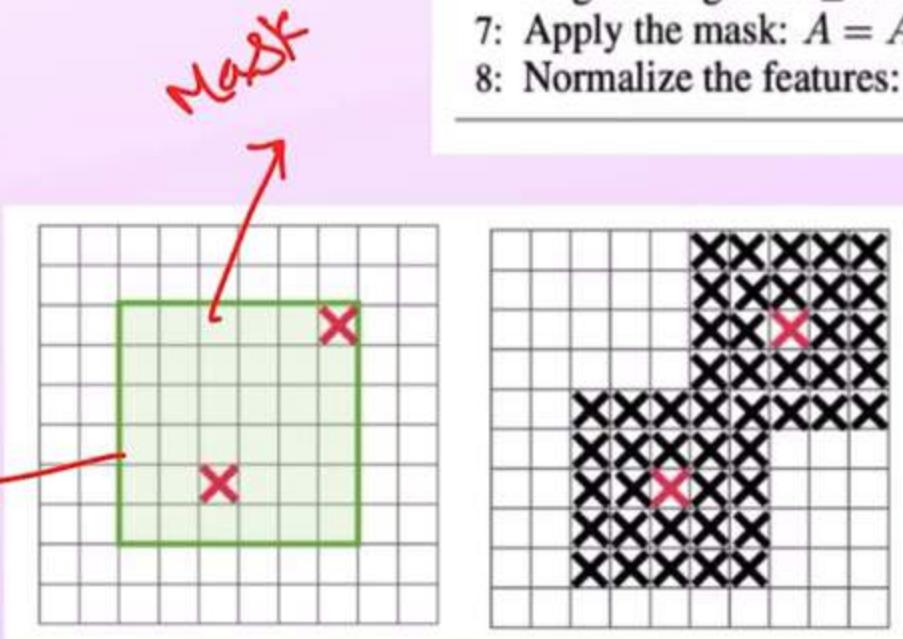
---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
- 



# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

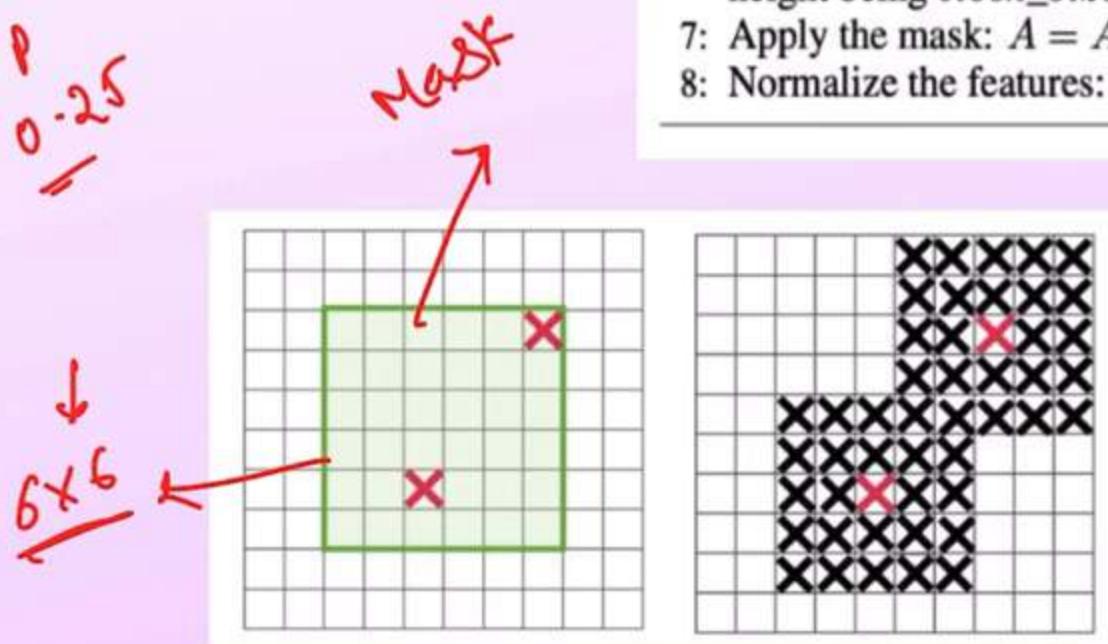
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

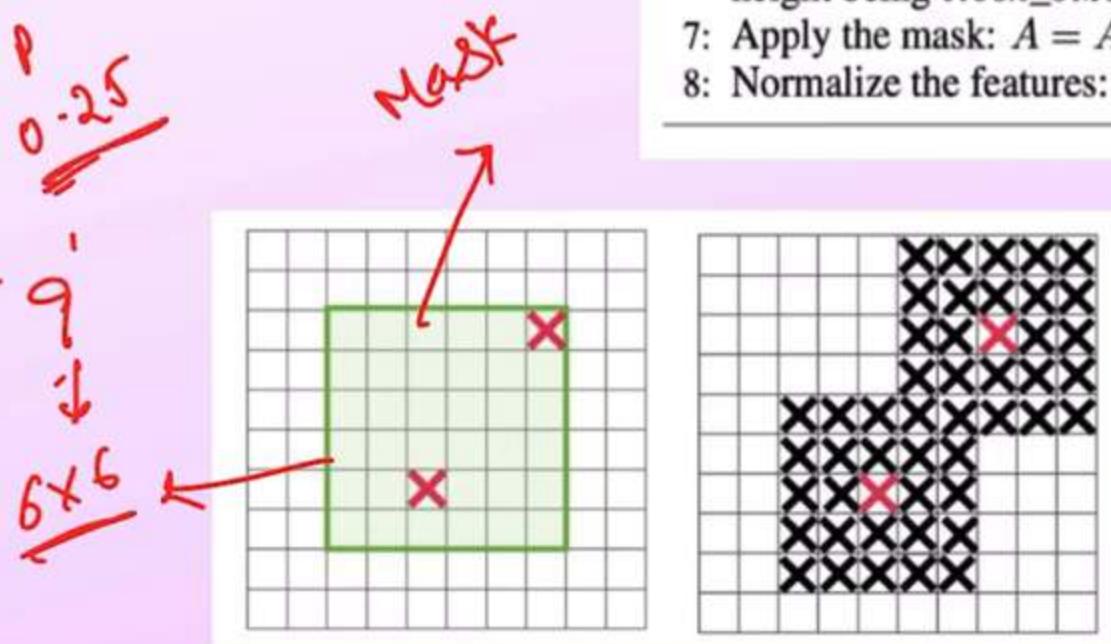
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

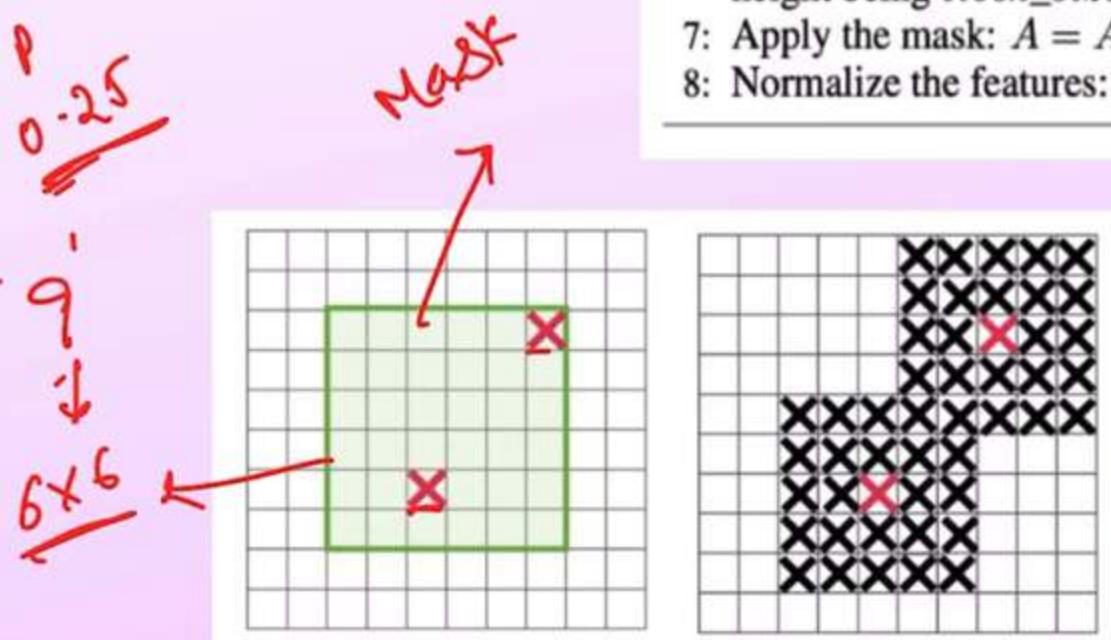
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

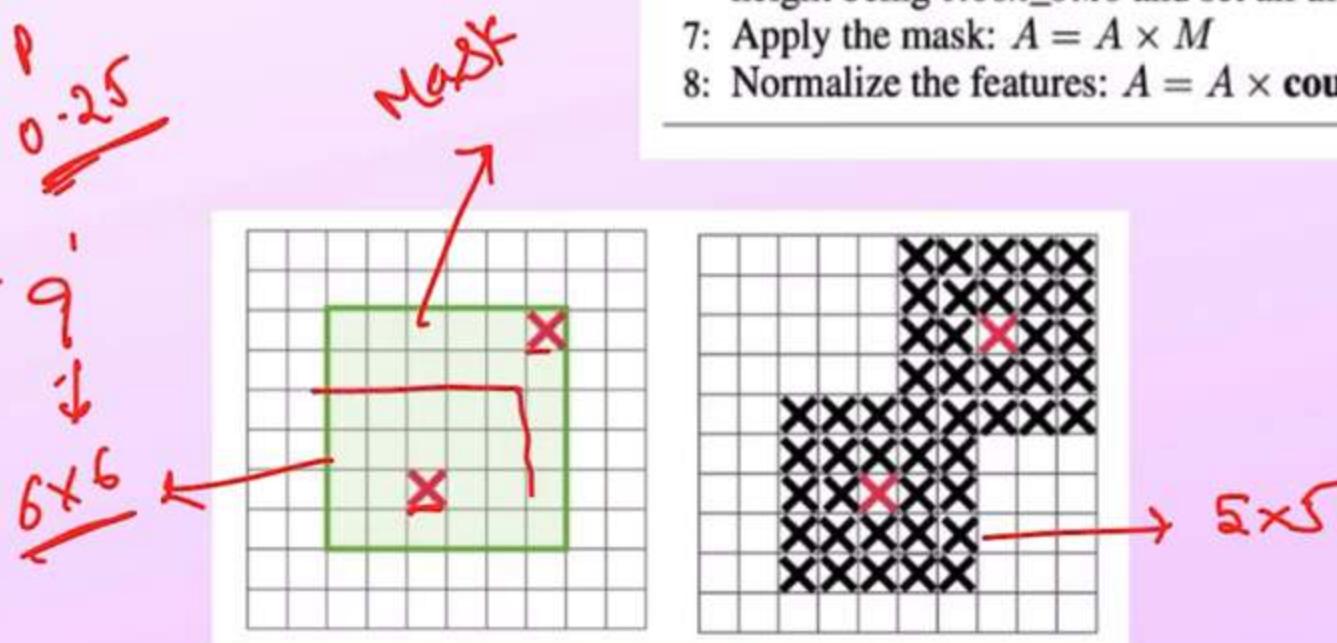
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability of Keeping



---

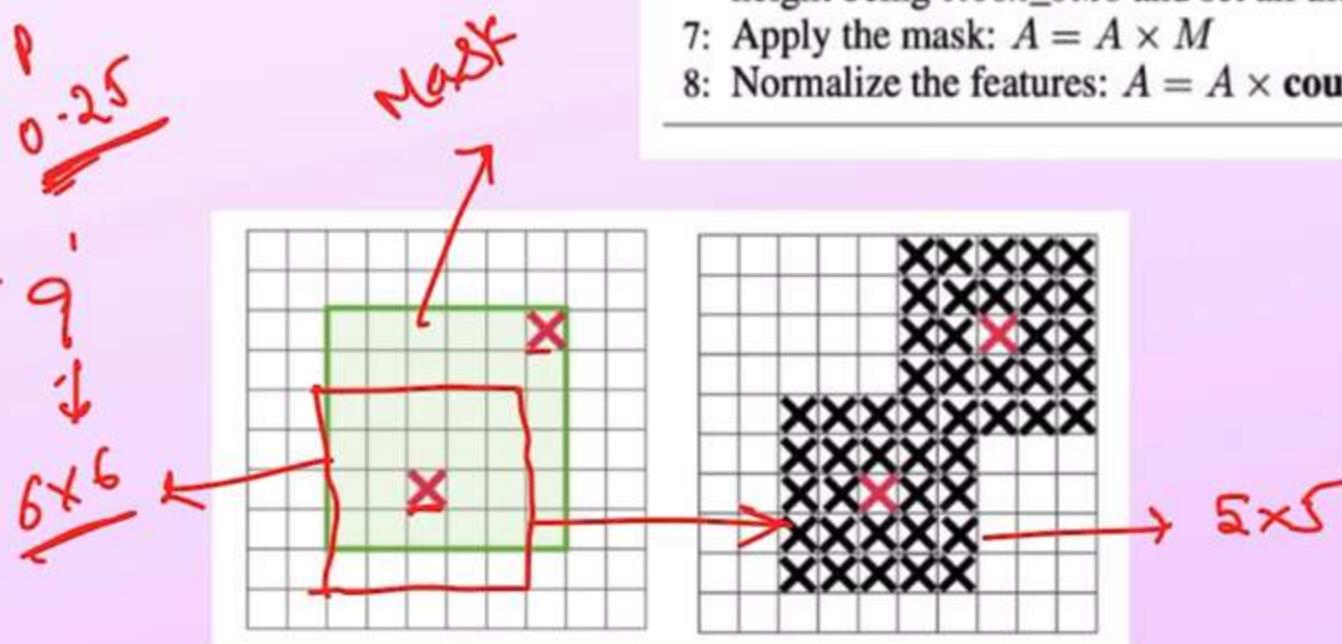
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

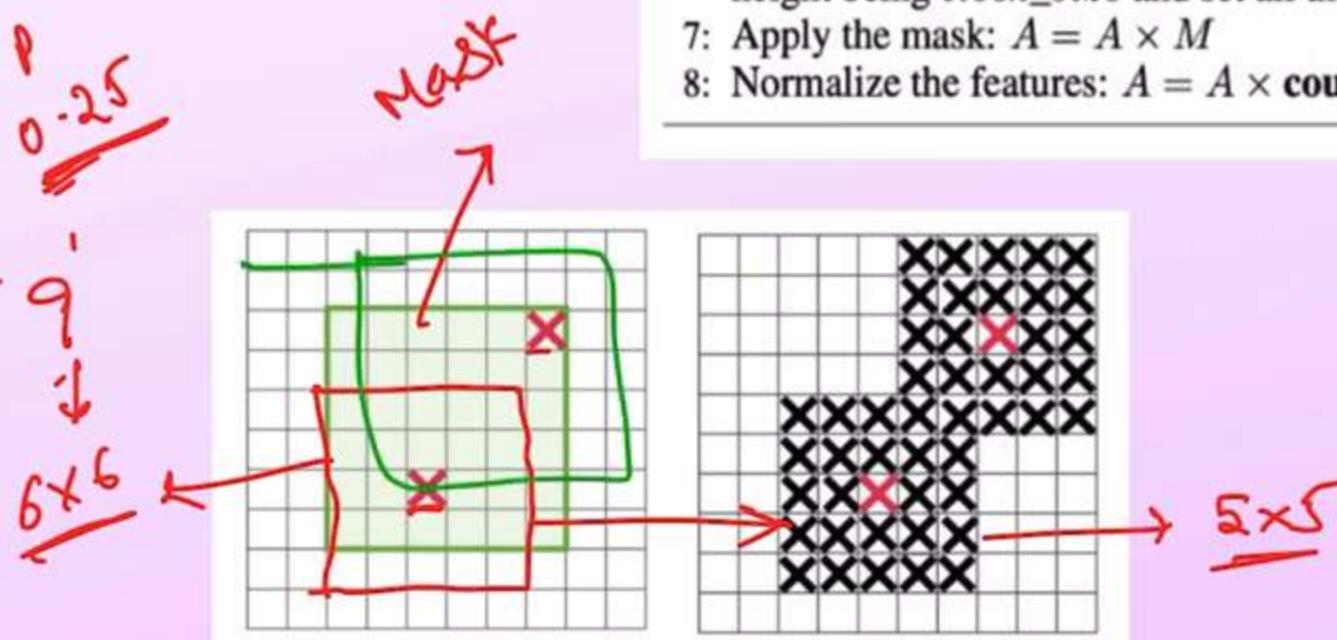
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

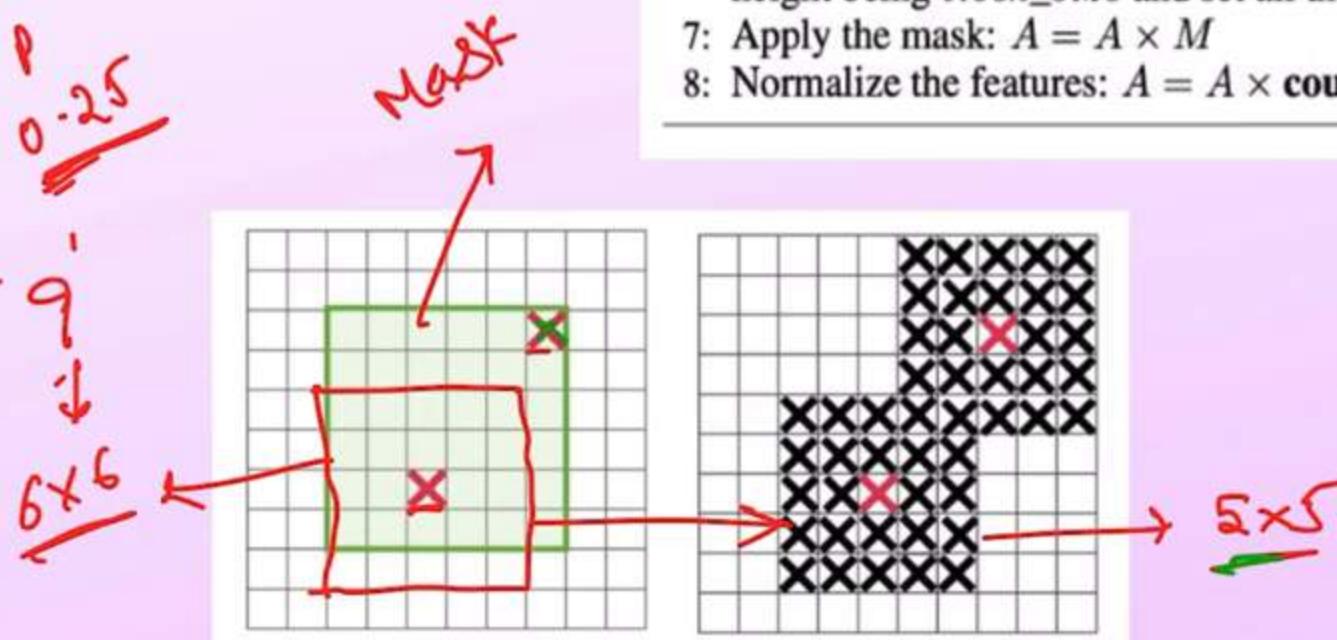
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

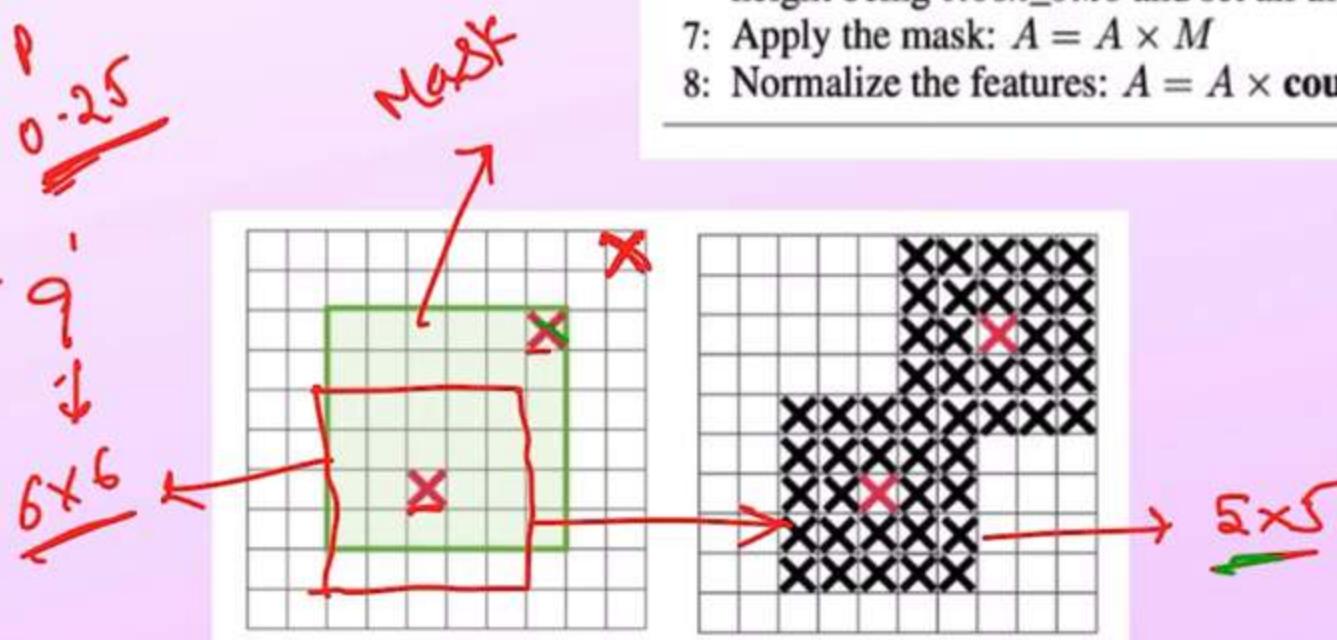
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

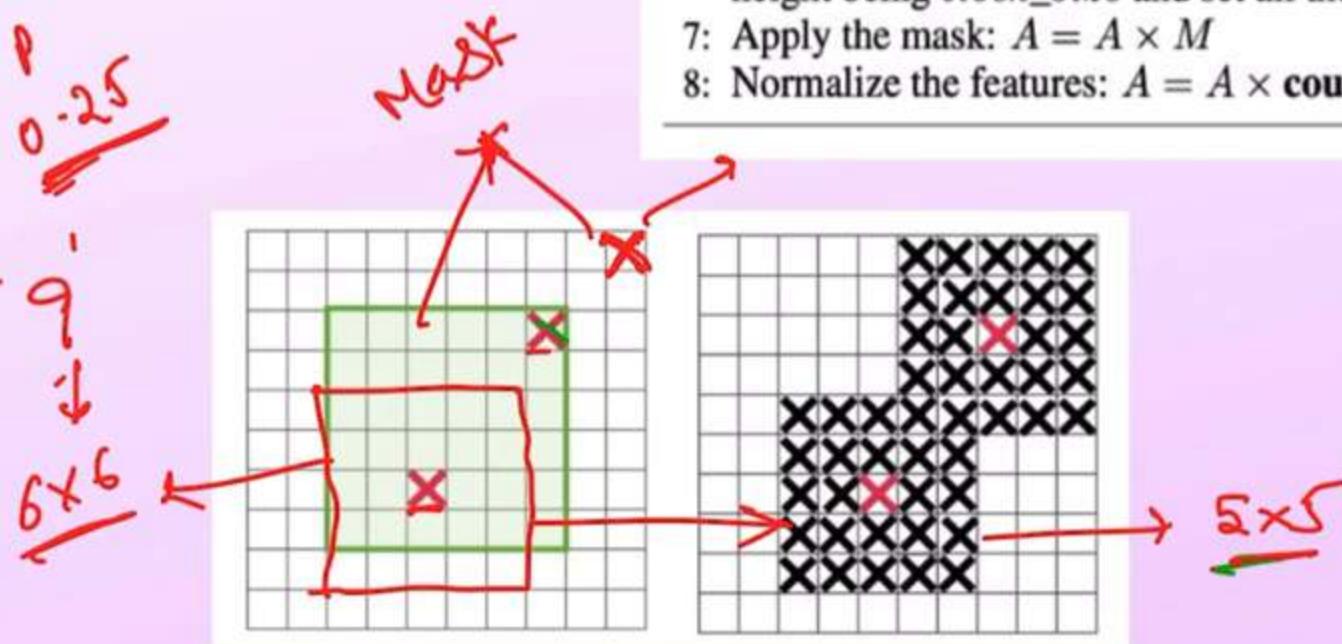
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability of Keeping



---

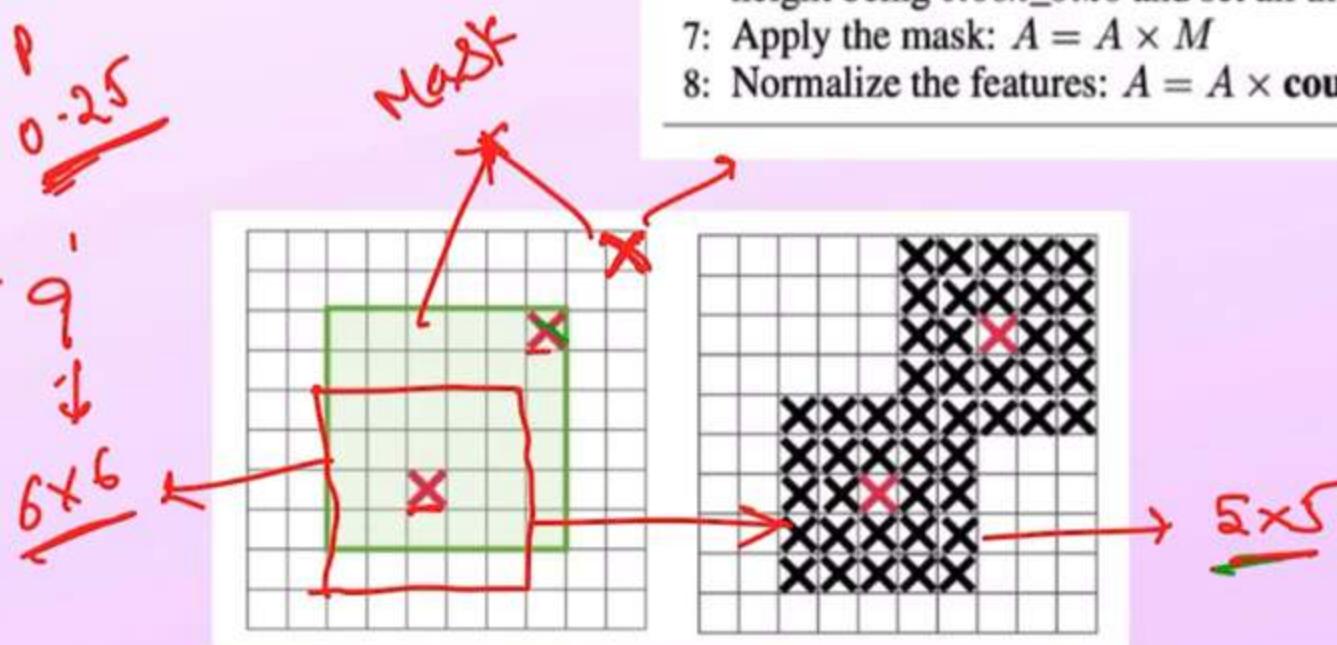
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:   **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

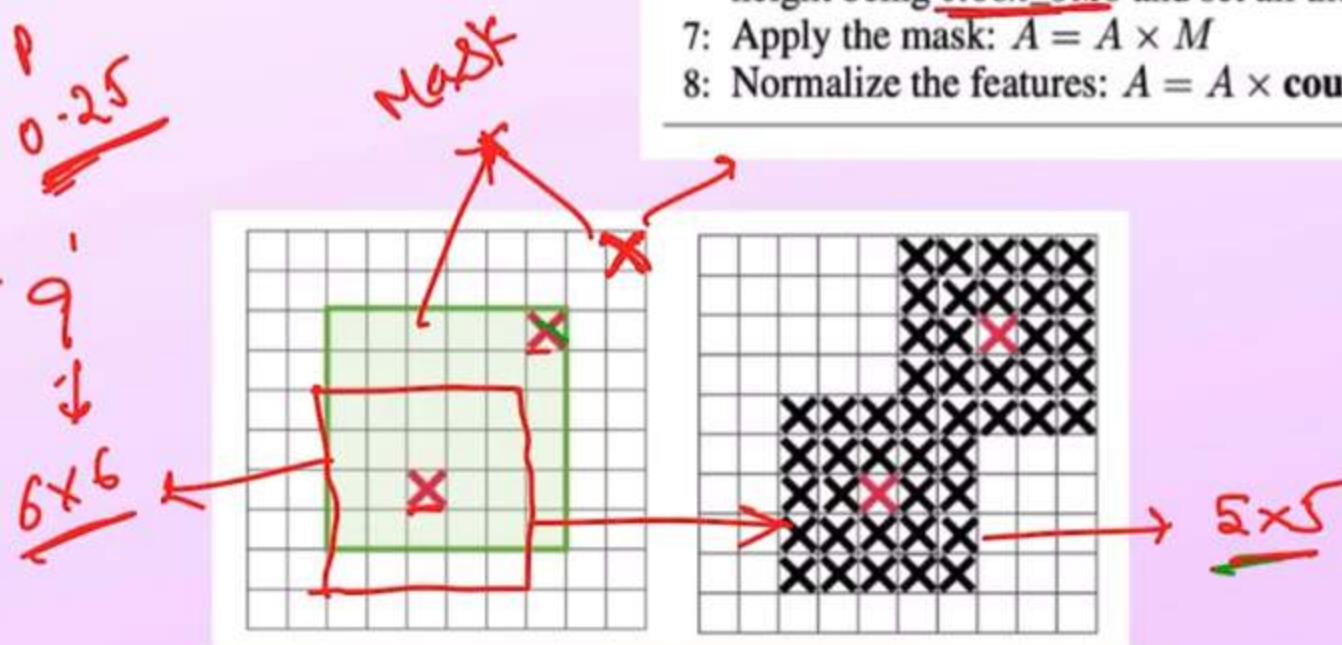
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

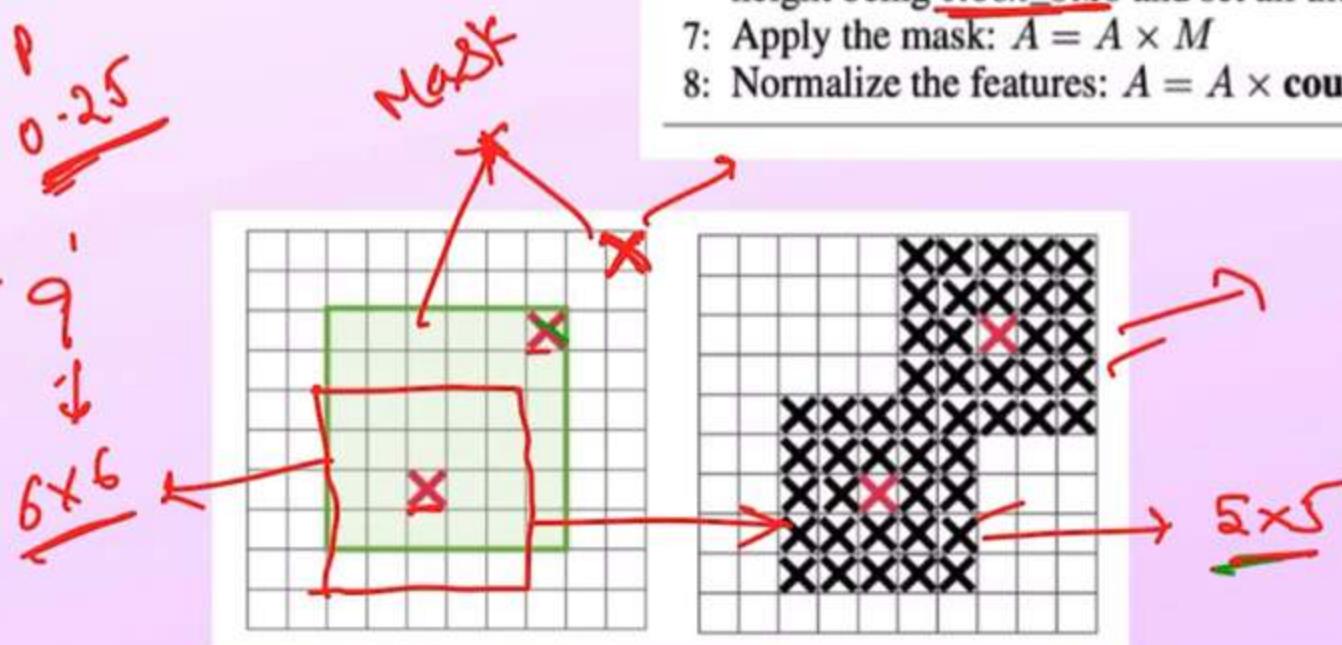
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

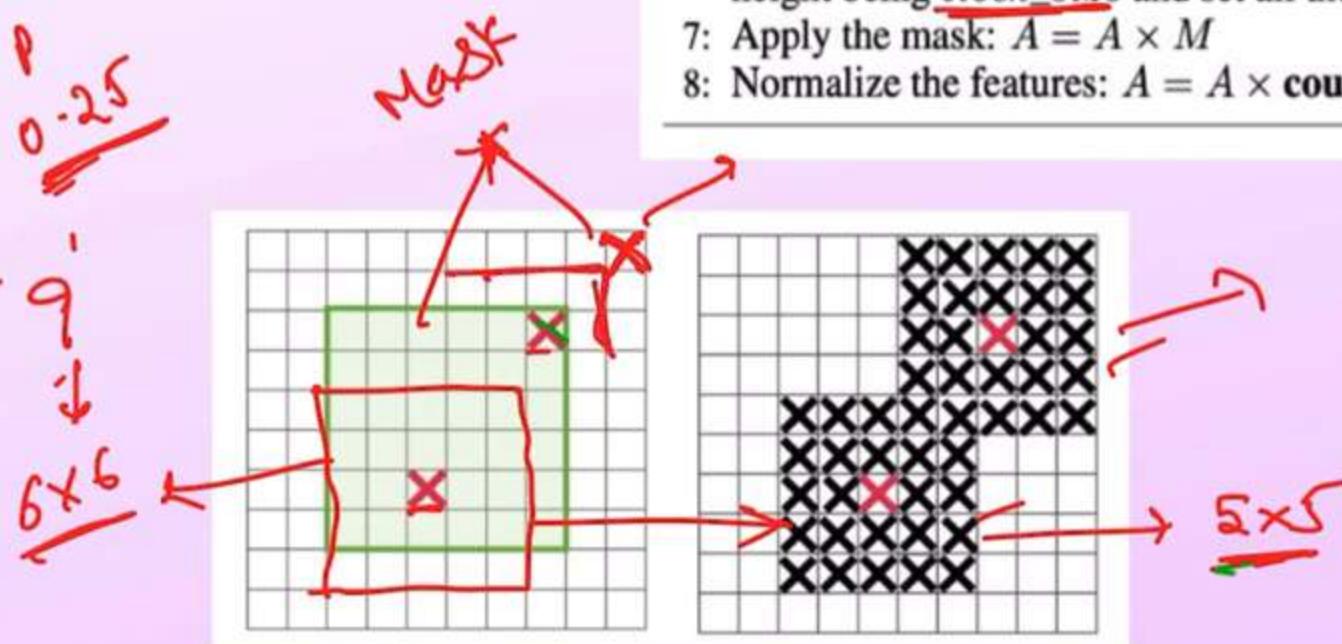
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

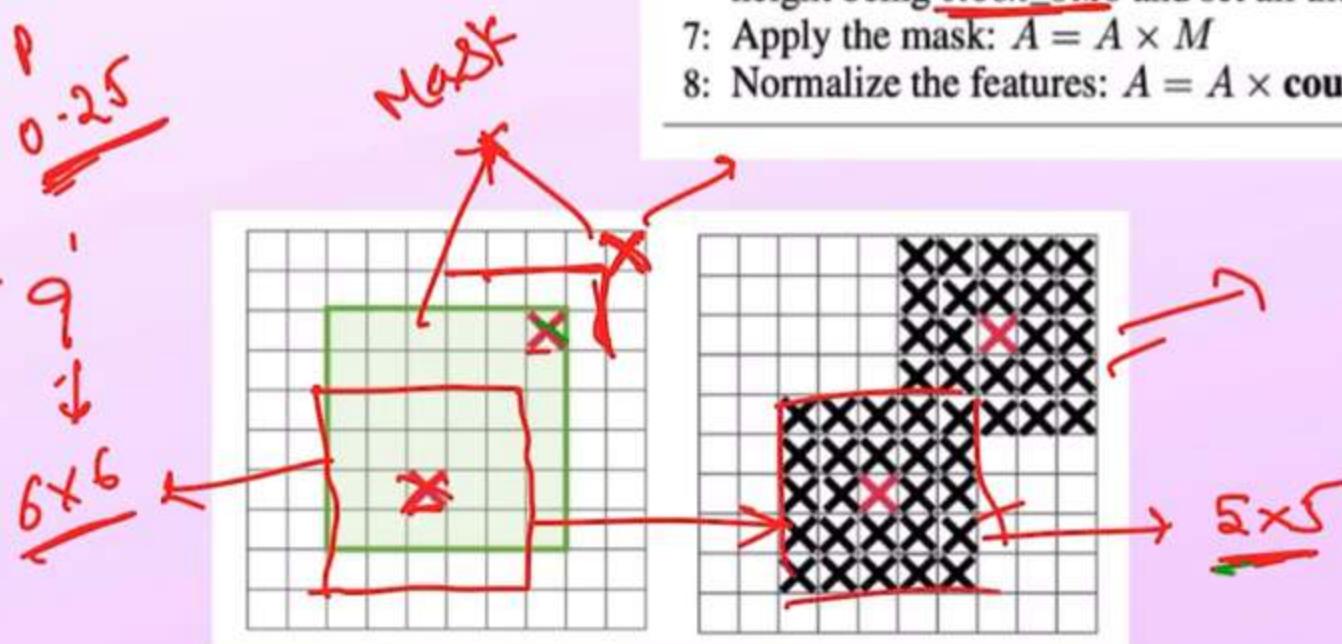
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- Block\_size
- $\gamma$  - Probability  
of Keeping



---

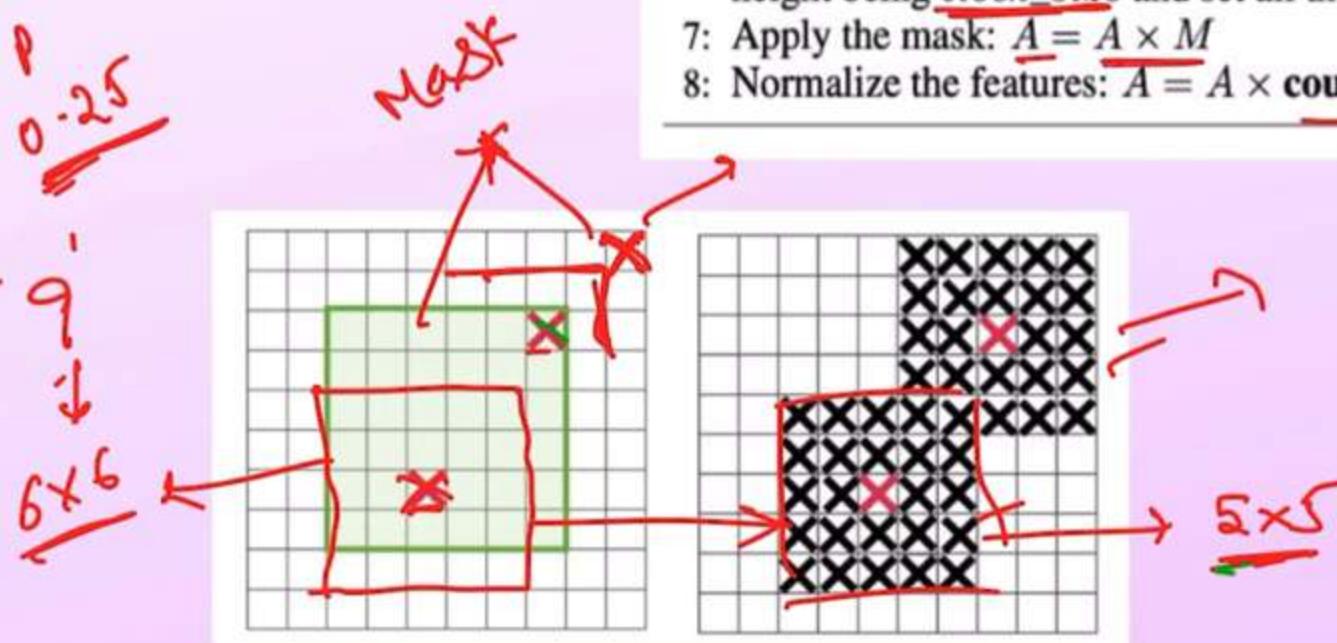
**Algorithm 1** DropBlock

---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ ,  $mode$
  - 2: **if**  $mode == \text{Inference}$  **then**
  - 3:     **return**  $A$
  - 4: **end if**
  - 5: Randomly sample mask  $M$ :  $M_{i,j} \sim \text{Bernoulli}(\gamma)$
  - 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being  $block\_size$  and set all the values of  $M$  in the square to be zero (see Figure 2).
  - 7: Apply the mask:  $A = A \times M$
  - 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count\_ones}(M)$
-

# Drop Block Regularization

- `Block_size`
- $\gamma$  - Probability  
of Keeping



---

**Algorithm 1** DropBlock

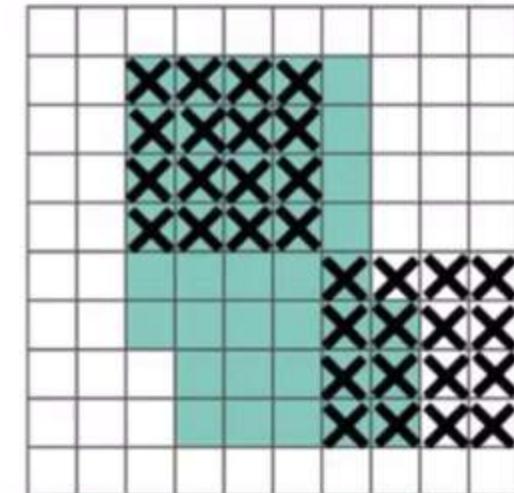
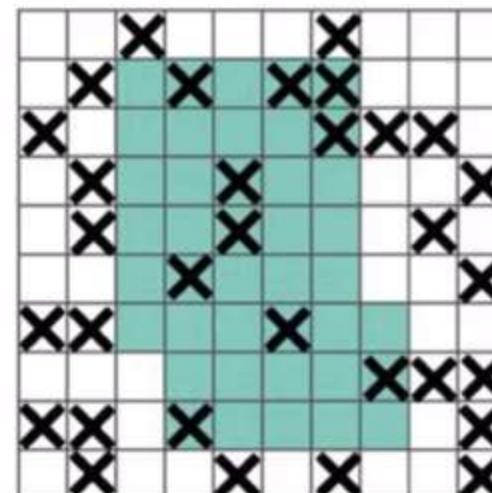
---

- 1: **Input:** output activations of a layer ( $A$ ),  $block\_size$ ,  $\gamma$ , mode
- 2: **if** mode == Inference **then**
- 3:   return  $A$
- 4: **end if**
- 5: Randomly sample mask  $M$ :  $M_{i,j} \sim Bernoulli(\gamma)$
- 6: For each zero position  $M_{i,j}$ , create a spatial square mask with the center being  $M_{i,j}$ , the width, height being block\_size and set all the values of  $M$  in the square to be zero (see Figure 2).
- 7: Apply the mask:  $A = A \times M$
- 8: Normalize the features:  $A = A \times \text{count}(M) / \text{count_ones}(M)$

---

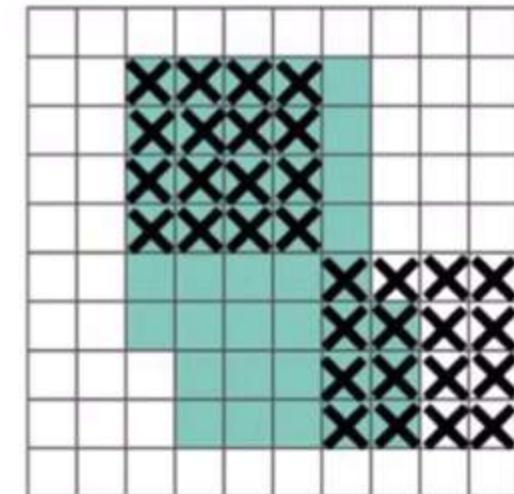
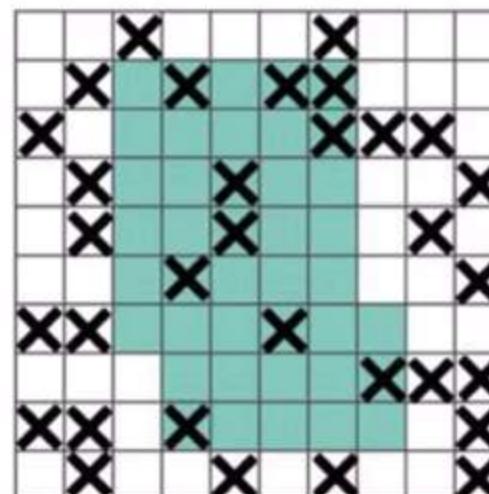
# Drop Block Regularization

- . ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- . RetinaNet (COCO) - 36.8% to 38.4%



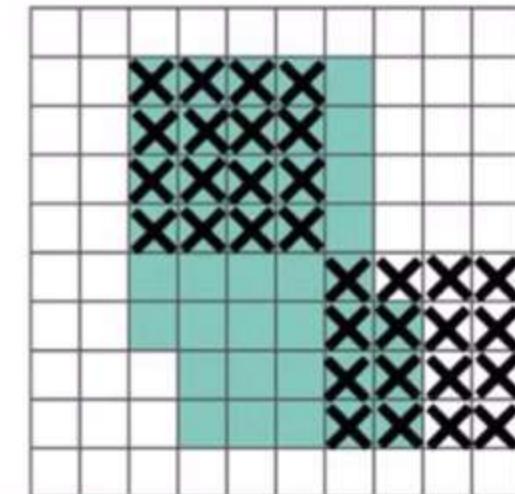
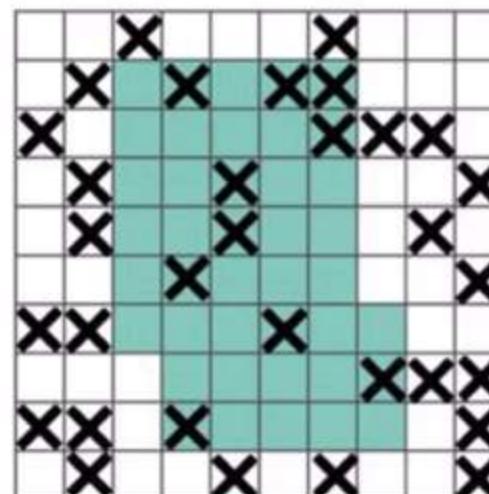
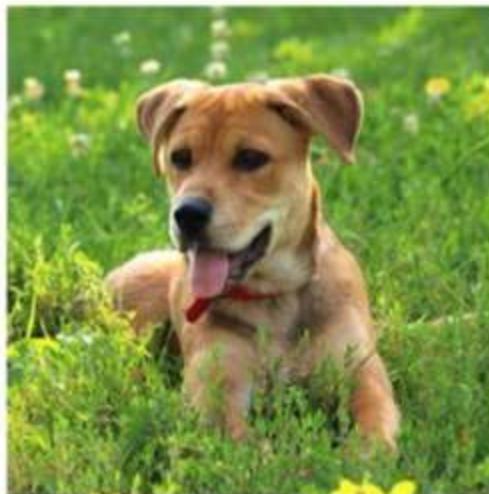
# Drop Block Regularization

- ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- RetinaNet (COCO) - 36.8% to 38.4%



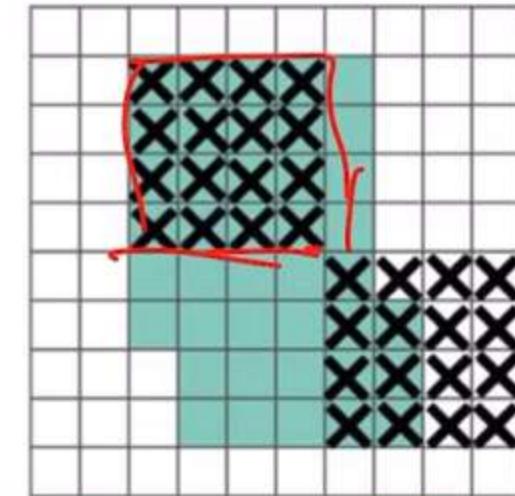
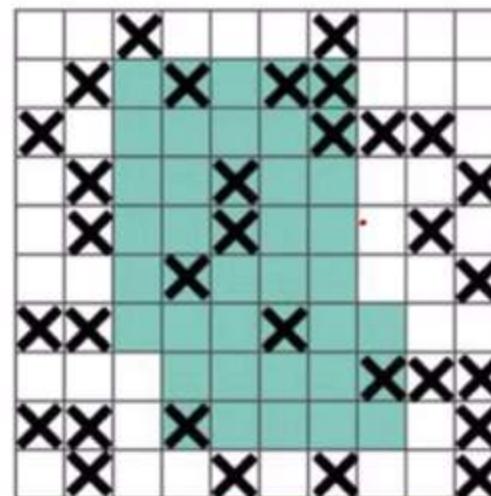
# Drop Block Regularization

- ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- RetinaNet (COCO) - 36.8% to 38.4%



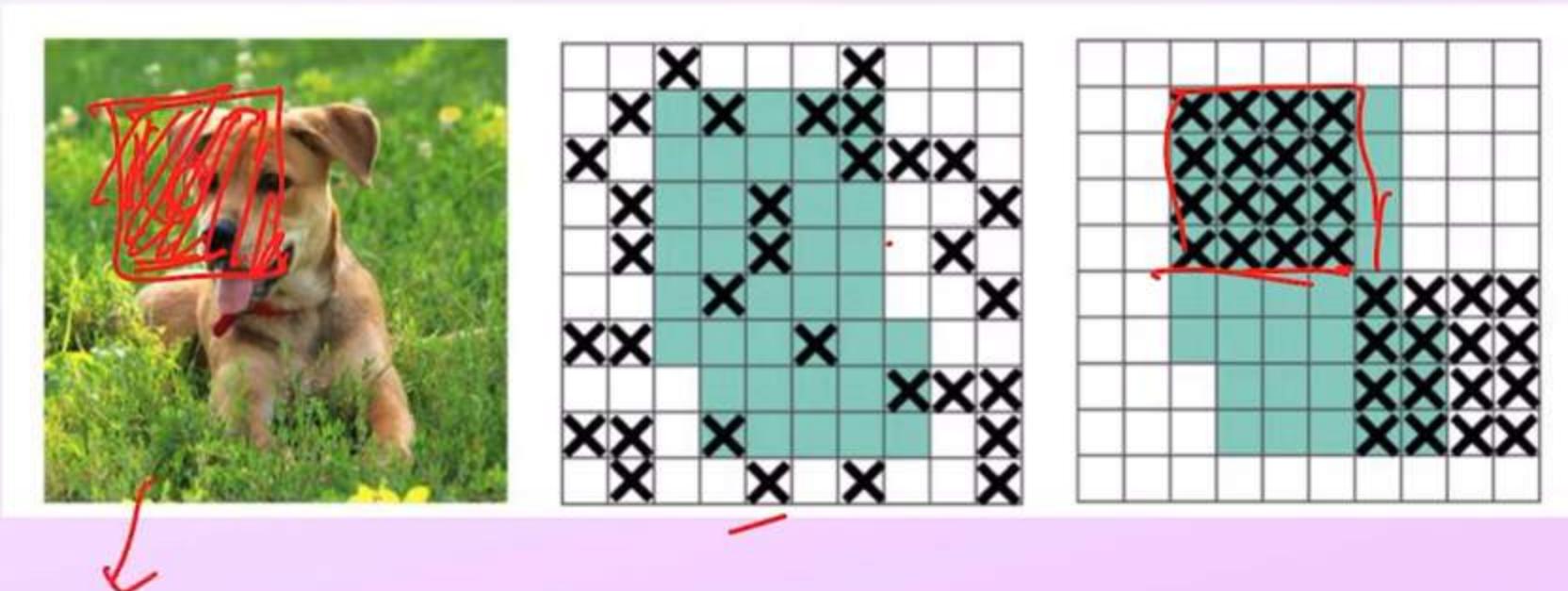
# Drop Block Regularization

- ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- RetinaNet (COCO) - 36.8% to 38.4%



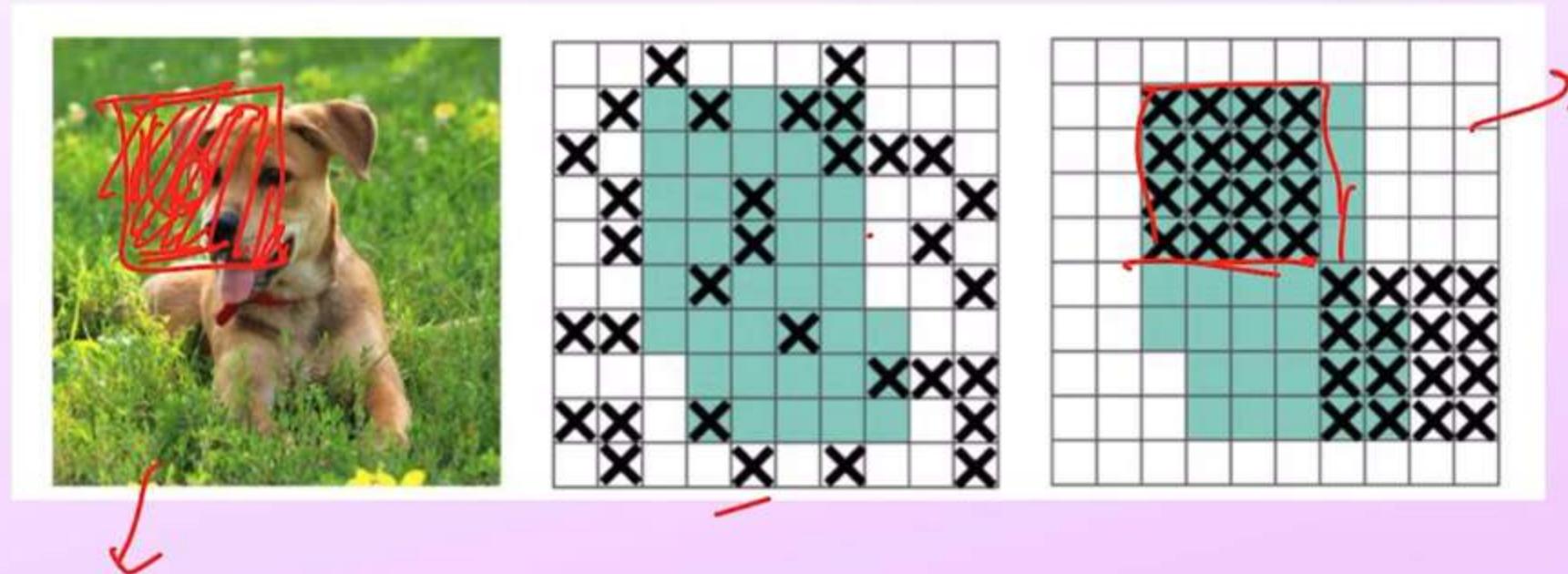
# Drop Block Regularization

- ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- RetinaNet (COCO) - 36.8% to 38.4%



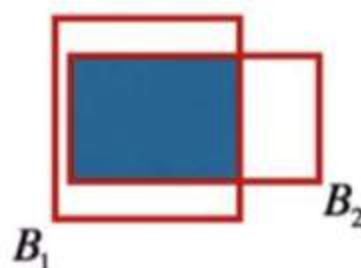
# Drop Block Regularization

- ResNet-50 (Imagenet) - Improved accuracy from 76.51% to 78.13%
- RetinaNet (COCO) - 36.8% to 38.4%

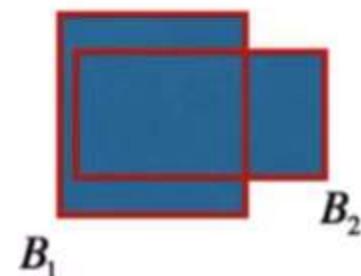


# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . The intersection area is highlighted with a red border, and the union area is highlighted with a larger red border, illustrating the components of the IoU formula.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

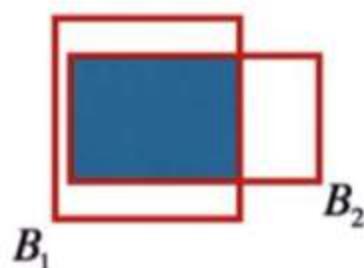
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

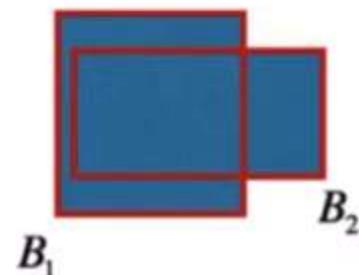
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection}}{\text{Union}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . A red rectangle surrounds both, representing their union. The area where the two blue rectangles overlap is shaded dark blue, representing their intersection. The ratio of the intersection area to the union area is calculated.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

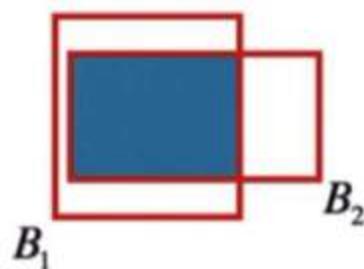
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

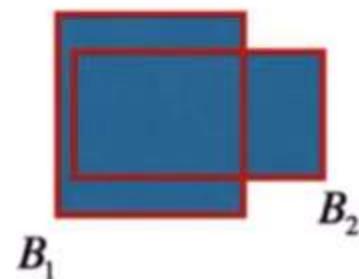
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . The intersection area is shaded dark blue and outlined in red. The union area is shaded blue and outlined in red. The ratio of the intersection area to the union area is calculated as the IoU value.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

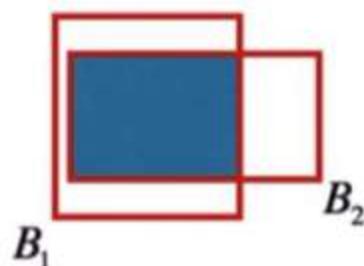
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

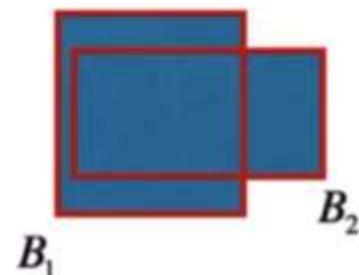
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . A red rectangle surrounds both the blue rectangles and the area where they overlap. The intersection area is highlighted in blue, and the union area is highlighted in red, illustrating the components of the IoU formula.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

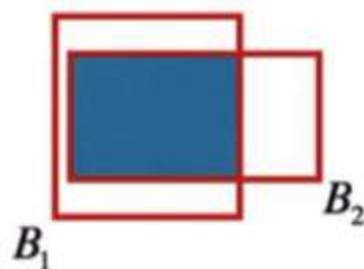
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

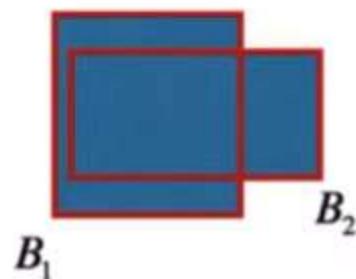
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

A diagram illustrating the IoU formula. It shows the intersection of two boxes,  $B_1$  and  $B_2$ , as a blue rectangle with a red border, and the union of the two boxes as a larger blue rectangle with a red border. The formula  $IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2}$  is shown above the diagram, where the numerator is the intersection area and the denominator is the union area.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

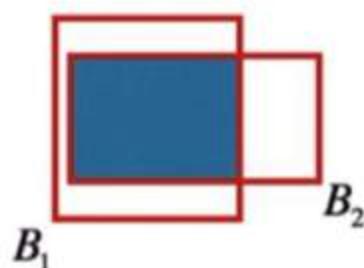
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

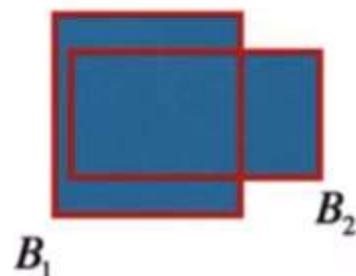
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Red Area}}{\text{Total Red Area}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . The intersection area is highlighted in red, and the union area is also highlighted in red, illustrating the components of the IoU formula.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

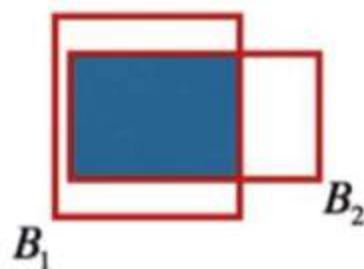
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

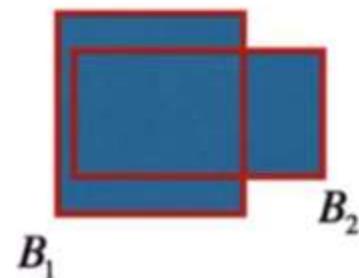
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection}}{\text{Union}}$$

A diagram showing two overlapping blue rectangles labeled  $B_1$  and  $B_2$ . The intersection area is shaded in dark blue, and the union area is shaded in light blue. The ratio of the intersection area to the union area is calculated to determine the IoU.

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

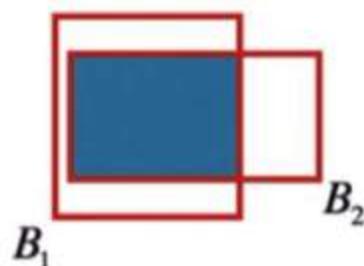
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

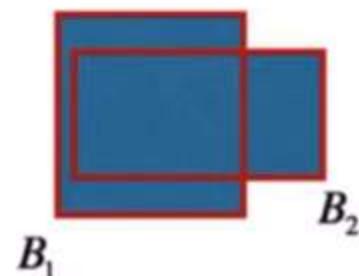
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

# IOU Loss

Intersection



Union



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

A diagram illustrating the formula for IoU. It shows two boxes,  $B_1$  and  $B_2$ , with their intersection shaded blue. The union of the two boxes is shown as a larger blue rectangle. A red arrow points from the formula to this diagram.

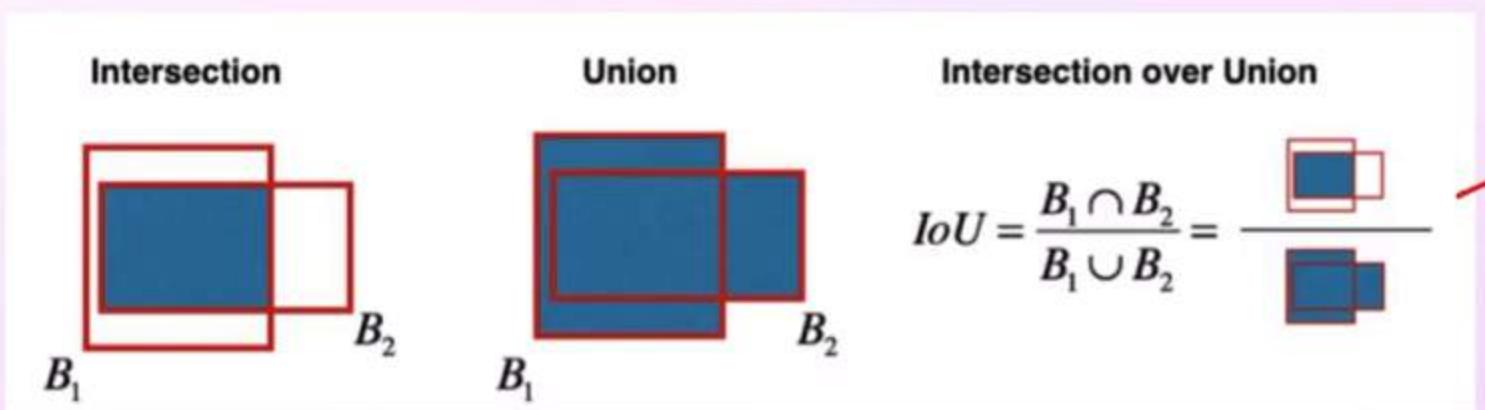
$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}} \rightarrow 1 - \underline{IoU}$$

# IOU Loss



1 → perfect overlap  
0 → no overlap

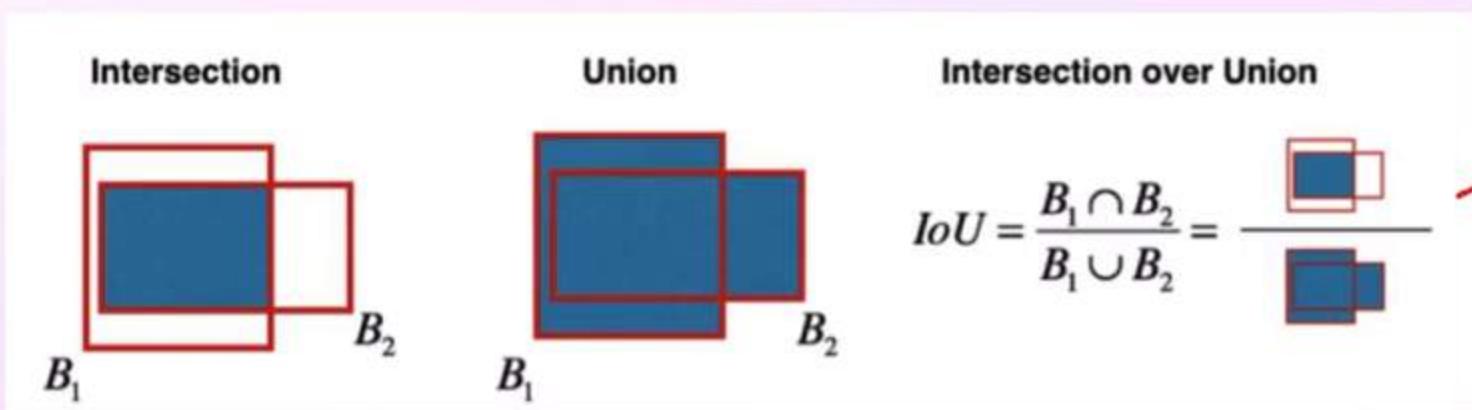
$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}} \rightarrow 1 - \underline{IoU}$$

# IOU Loss



1 → perfect overlap  
0 → no overlap

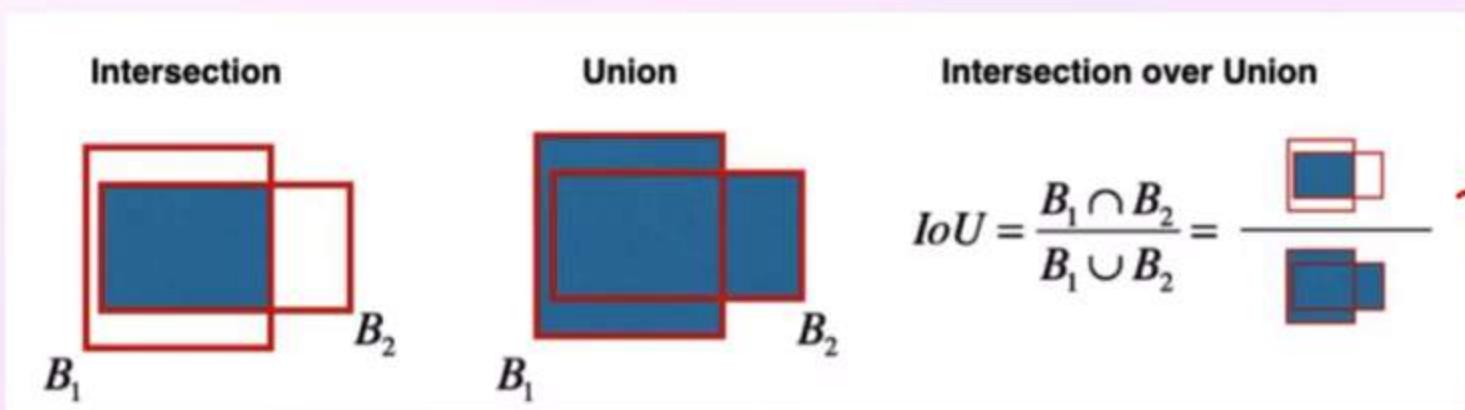
$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}} \rightarrow 1 - \underline{IoU}$$

# IOU Loss



IoU ↑  
1 → perfect overlap  
0 → no overlap

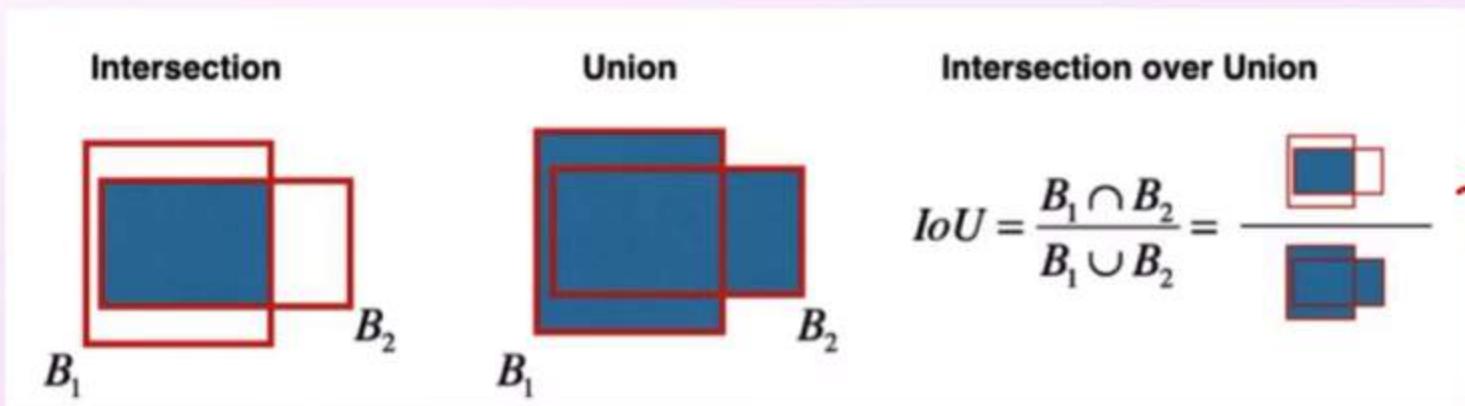
$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}} \rightarrow 1 - \underline{IoU}$$

# IOU Loss



IoU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

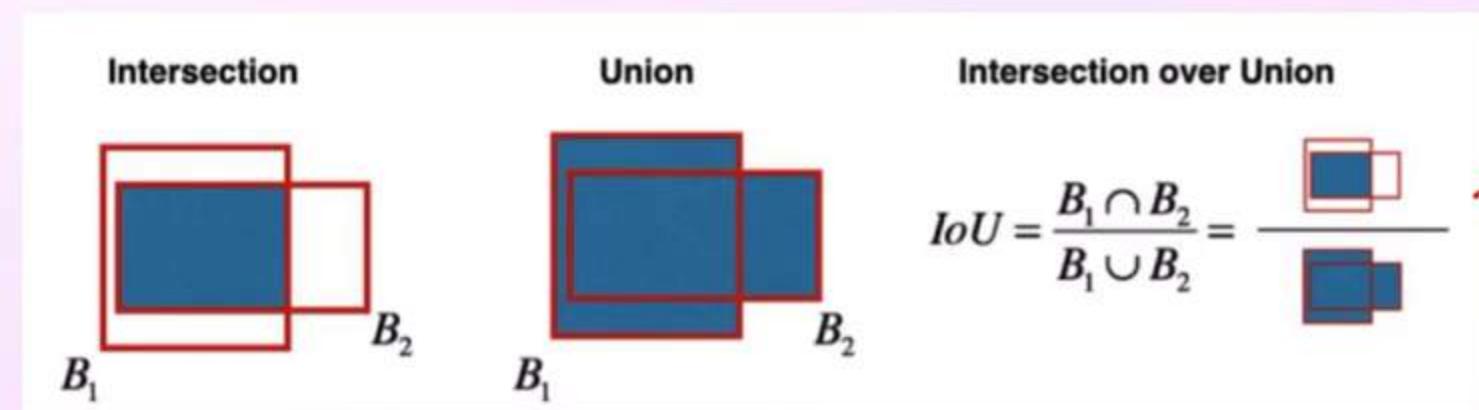
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}} \rightarrow \underline{1 - IoU}$$

Loss ↓

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

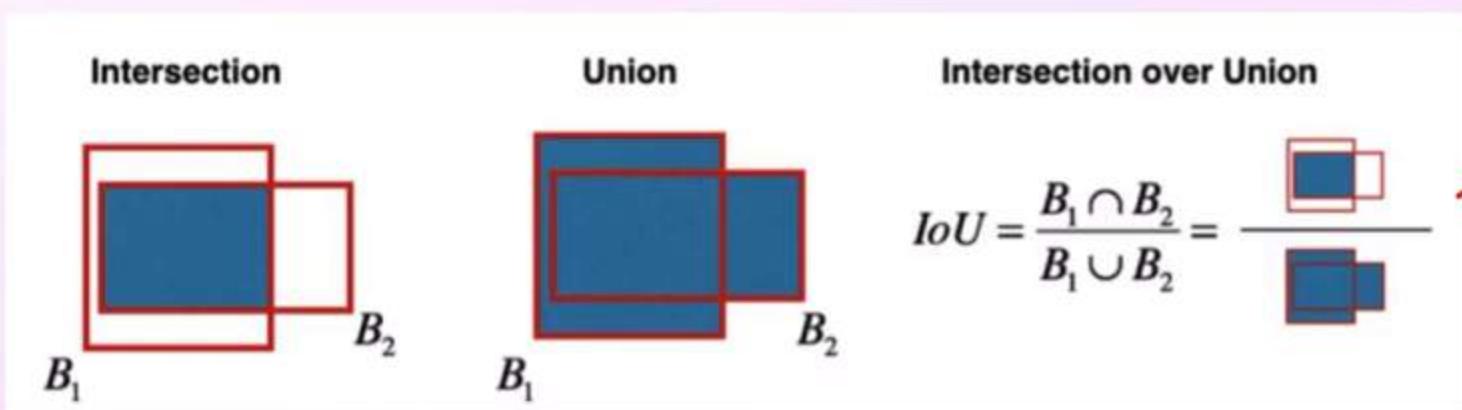
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

1 - IOU

Loss ↓

IoU  
0.7

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth  
 $B = (x, y, w, h)$  is the predicted box

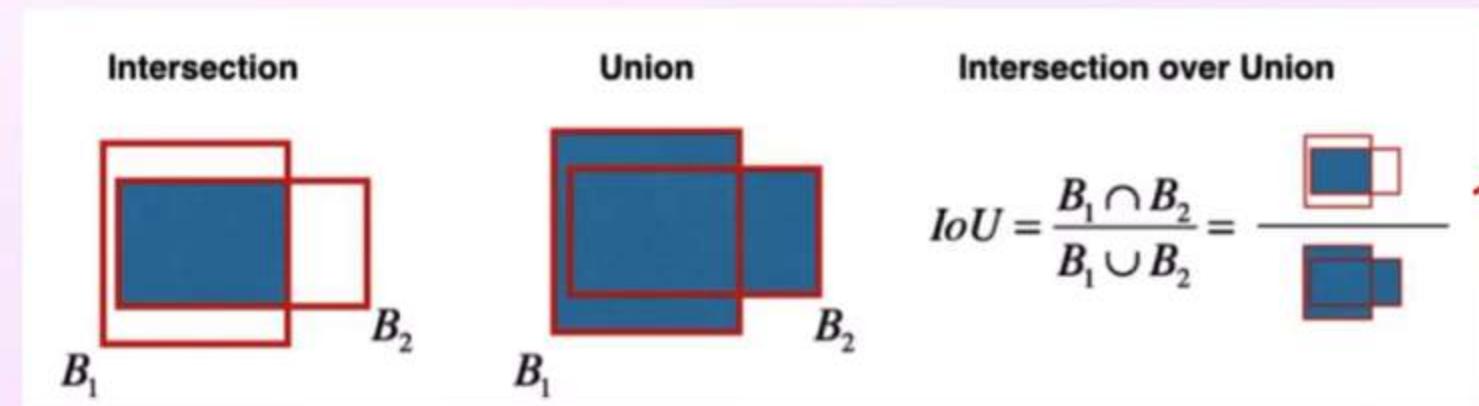
Loss ↓

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$\rightarrow 1 - IoU \rightarrow$   
 $\downarrow$   
 $\rightarrow 0.3 \downarrow$

IOU  
0.7

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

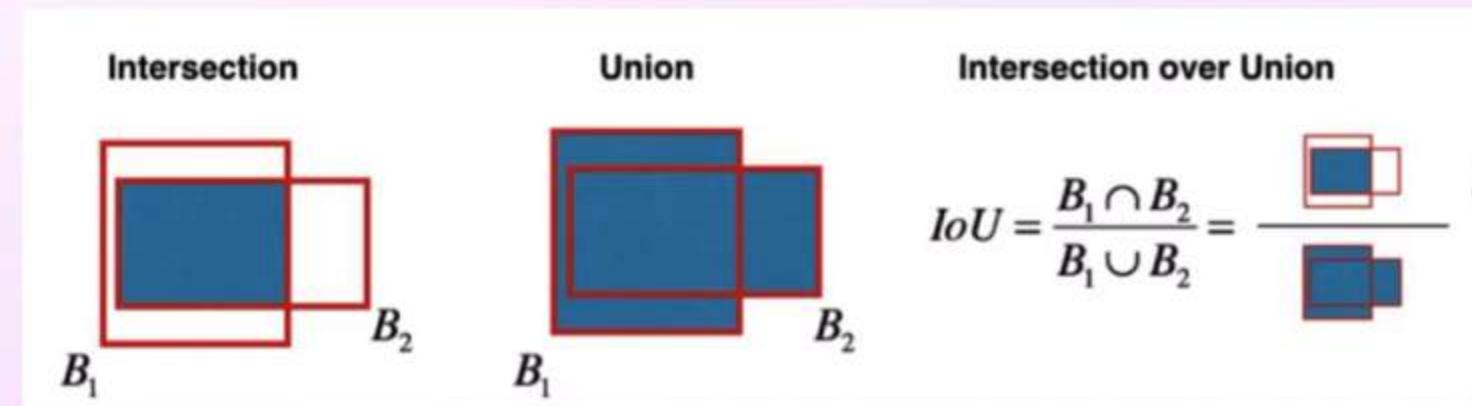
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Log<sub>s</sub> ↓  
 $\frac{IoU}{0.7}$   
 $1 - IoU \rightarrow$   
 $0.3 \downarrow$  non overlapping

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

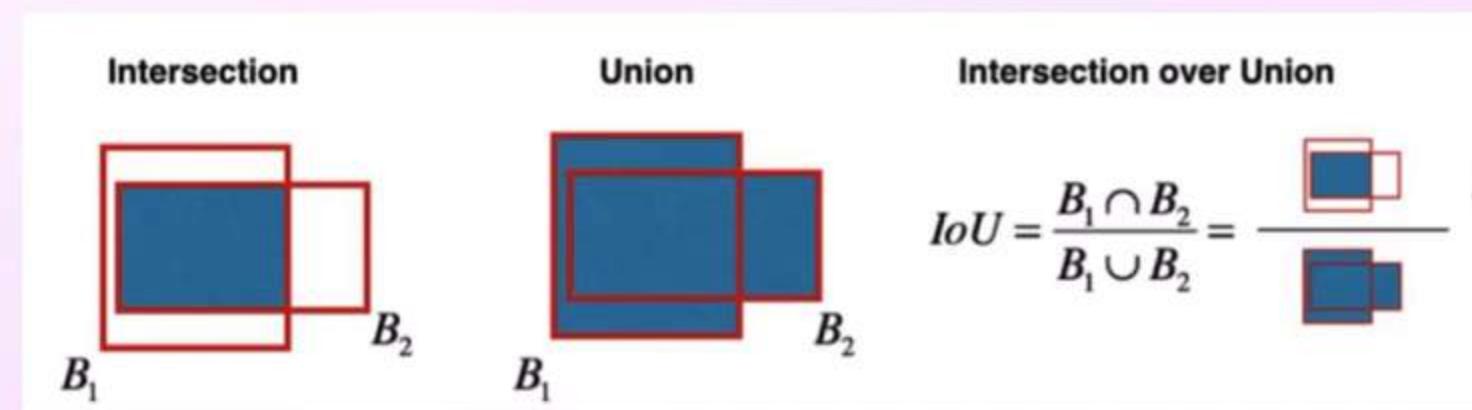
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Log ↓  
 $\rightarrow 1 - IoU \rightarrow$   
 $\rightarrow 0.3 \downarrow$  non overlapping region  
 $\frac{0.7}{0}$

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

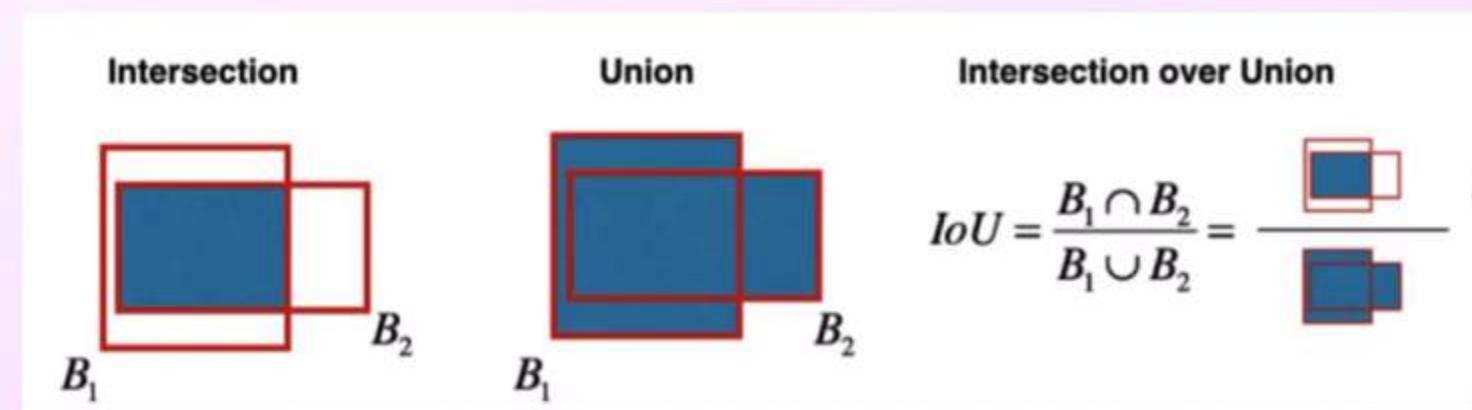
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth  
 $B = (x, y, w, h)$  is the predicted box

Loss ↓

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$1 - IoU$  → 0.7  
0.3 ↓ non overlapping region  
0: non overlapping region

# IOU Loss



IOU ↑  
1 → perfect overlap  
0 → no overlap

$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

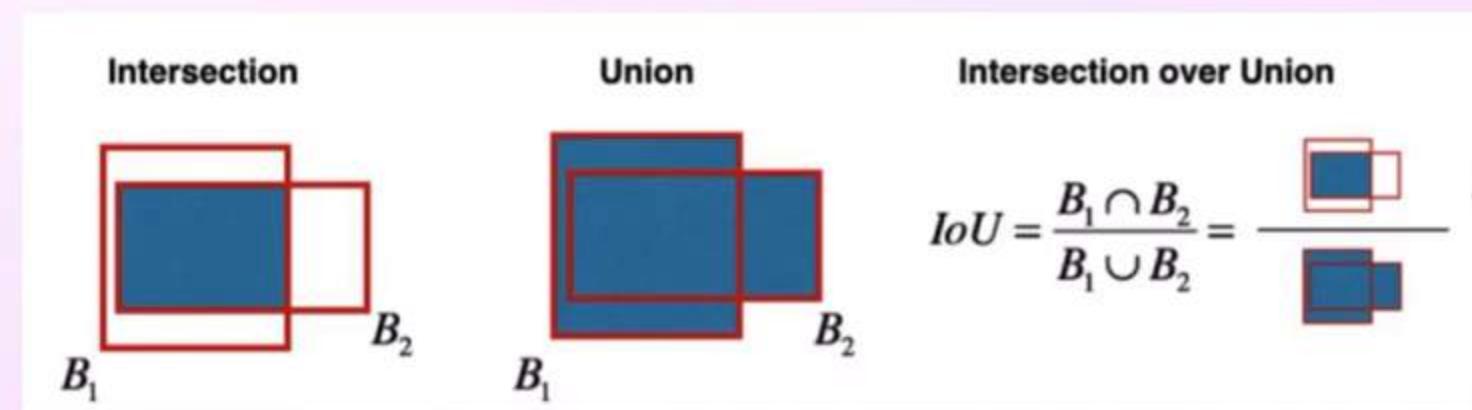
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Loss ↓  
 $1 - IoU \rightarrow$   
 $\frac{0.7}{0.3} \downarrow$  non overlapping region  
 $0.$

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

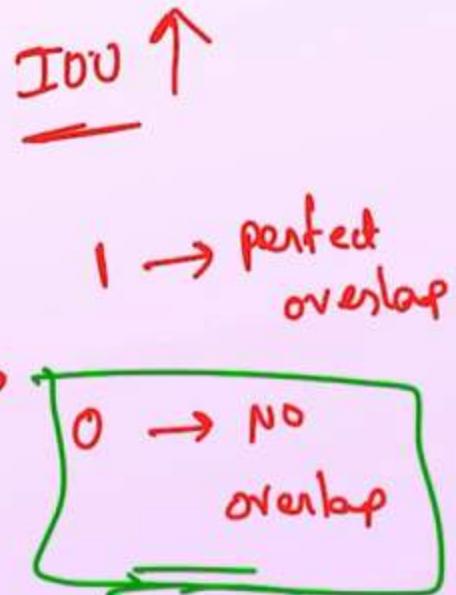
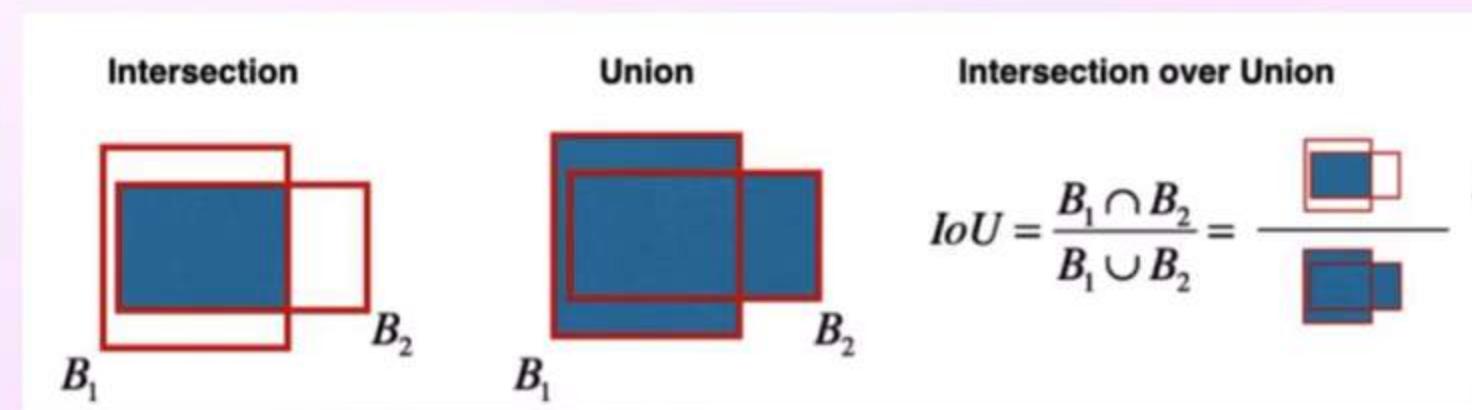
$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

→  $1 - IoU$   
→  $0.7$   
→  $0.3$   
↓ non overlapping region

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Loss ↓

IOU

0.7

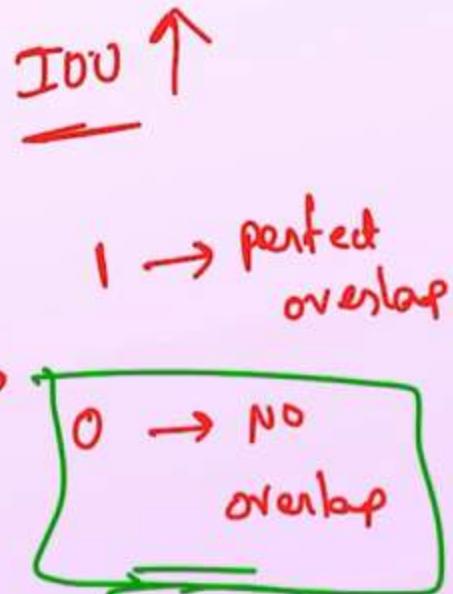
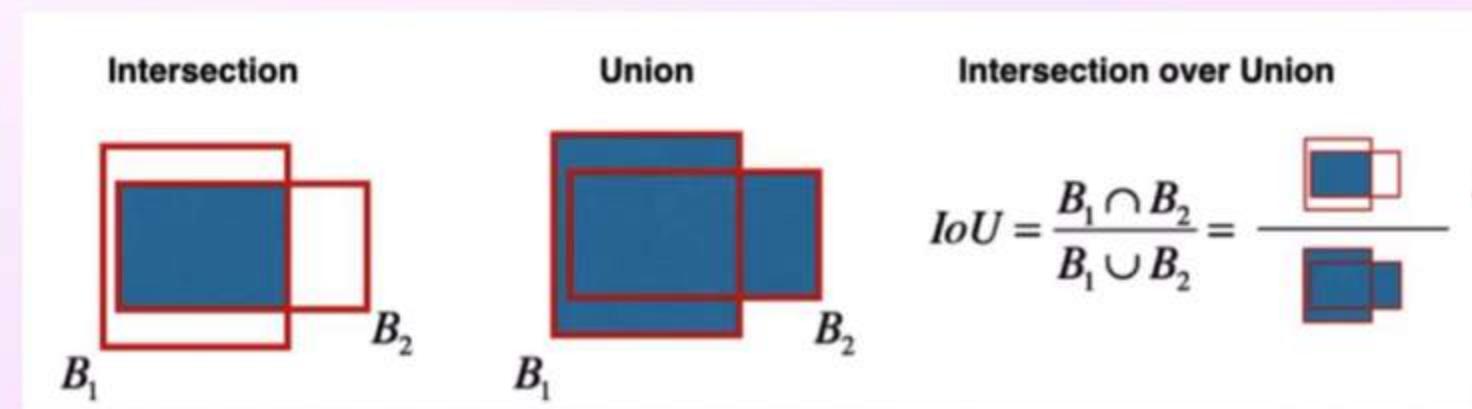
0.3

0.0

non overlapping region

GT  
OI

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

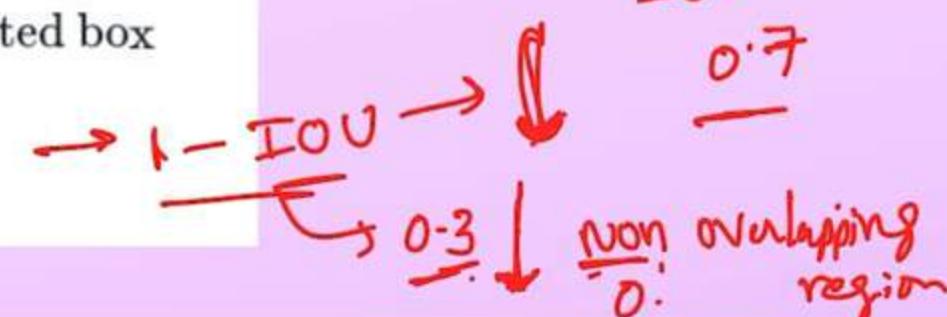
Log ↓

IOU ↓

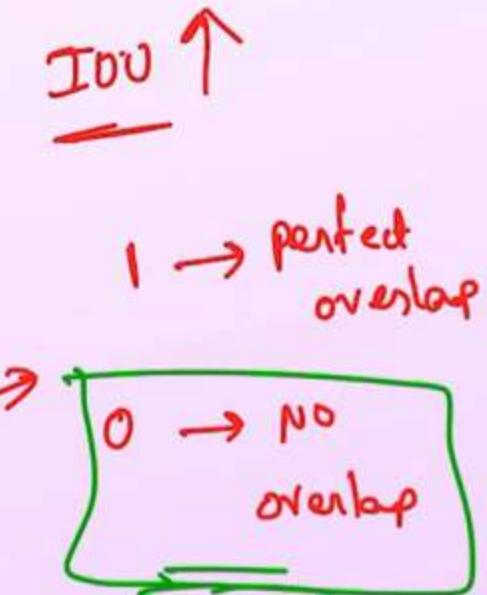
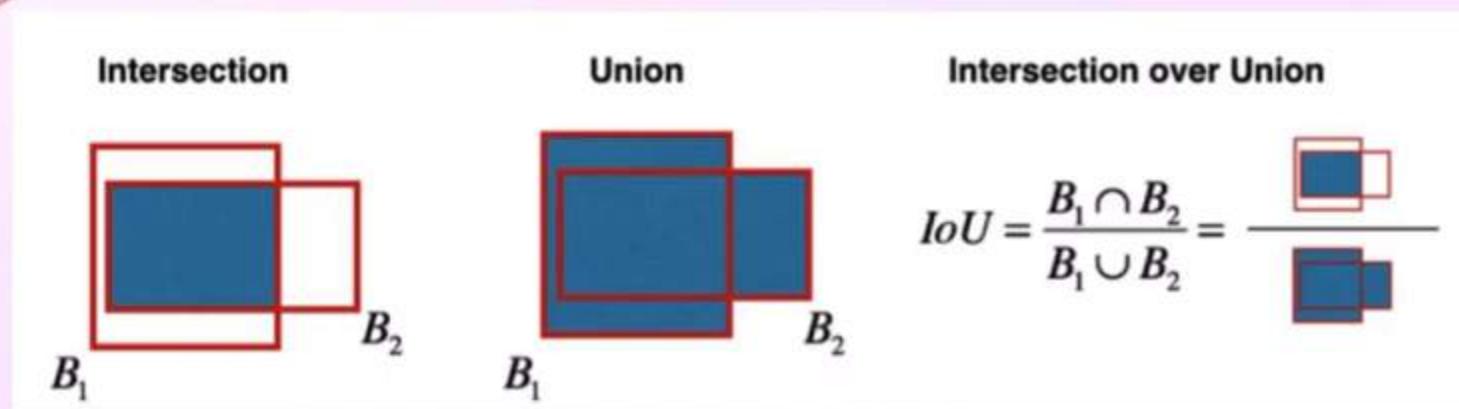
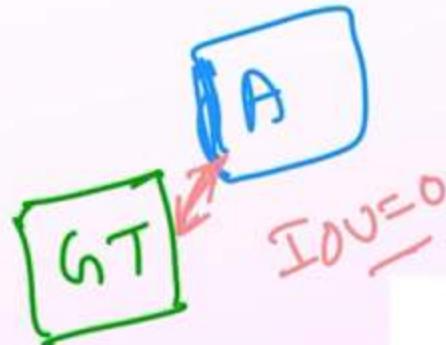
0.7

0.3 ↓ non overlapping region

0.



# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

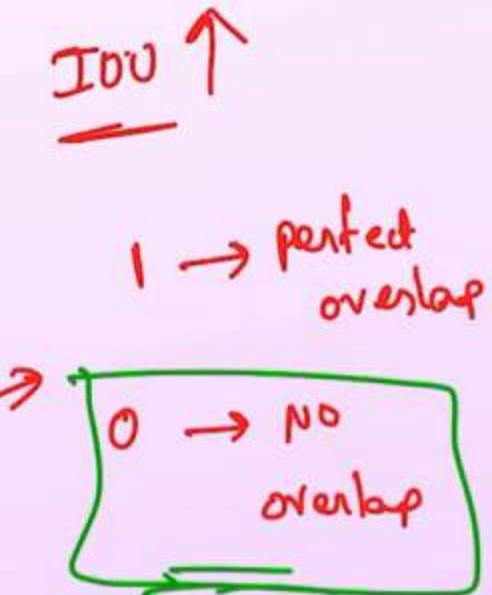
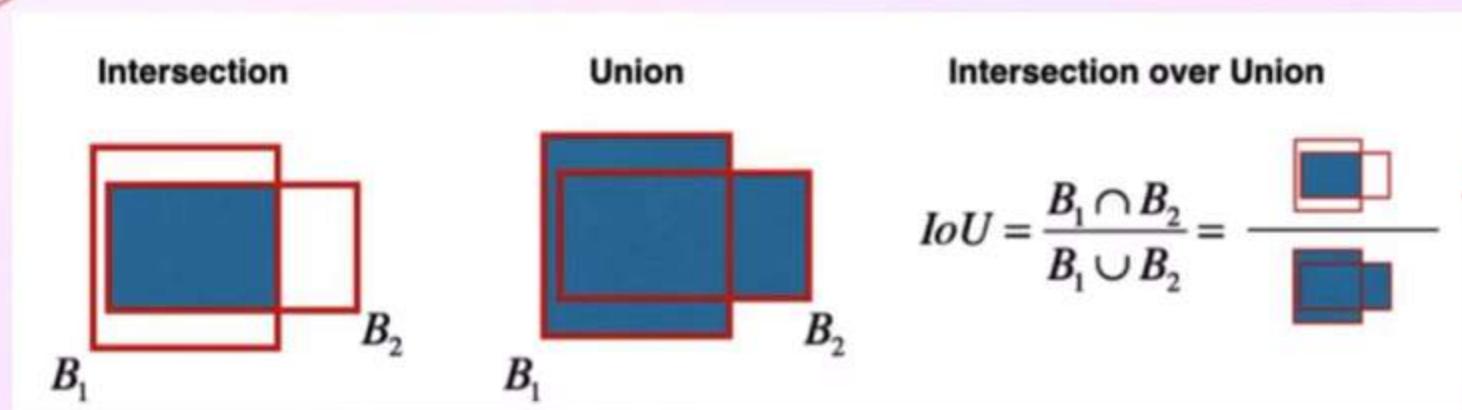
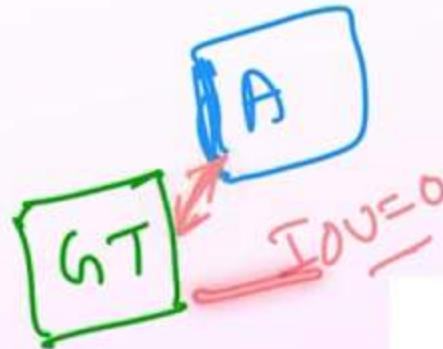
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Loss ↓

$$\frac{IoU}{0.7}$$

$\frac{0.3}{0.7}$  ↓ non overlapping region

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

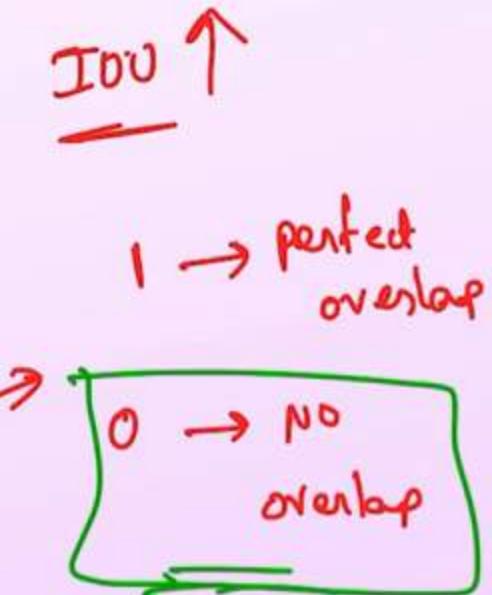
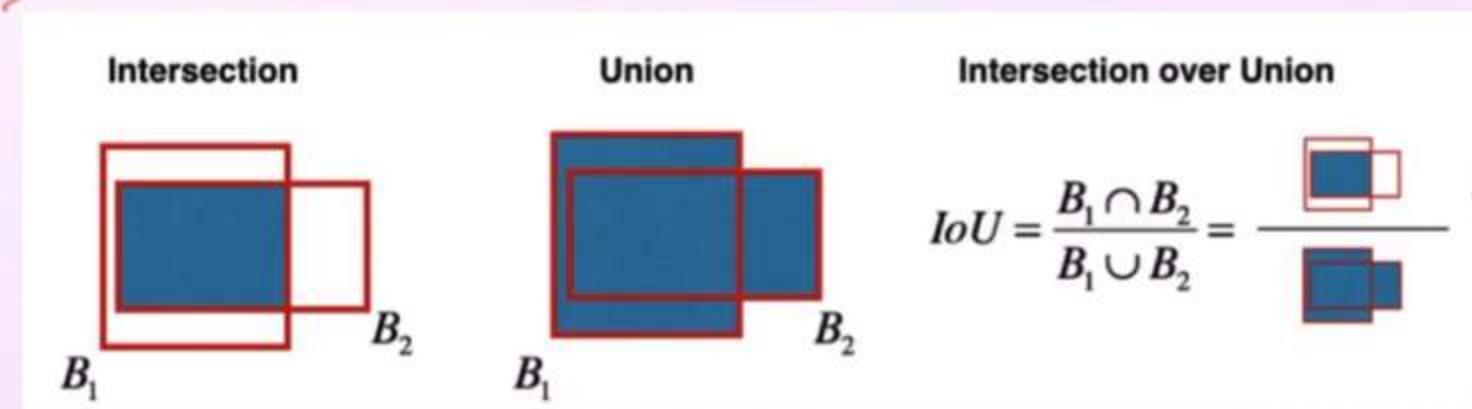
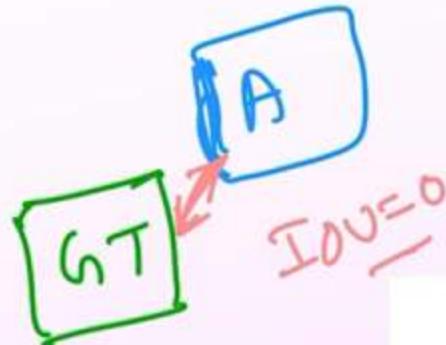
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Loss ↓

$$\frac{IoU}{0.7}$$

0.3 ↓ non overlapping region

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

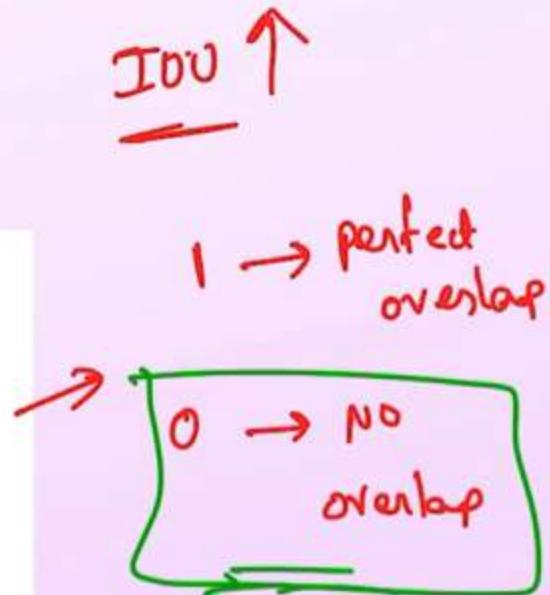
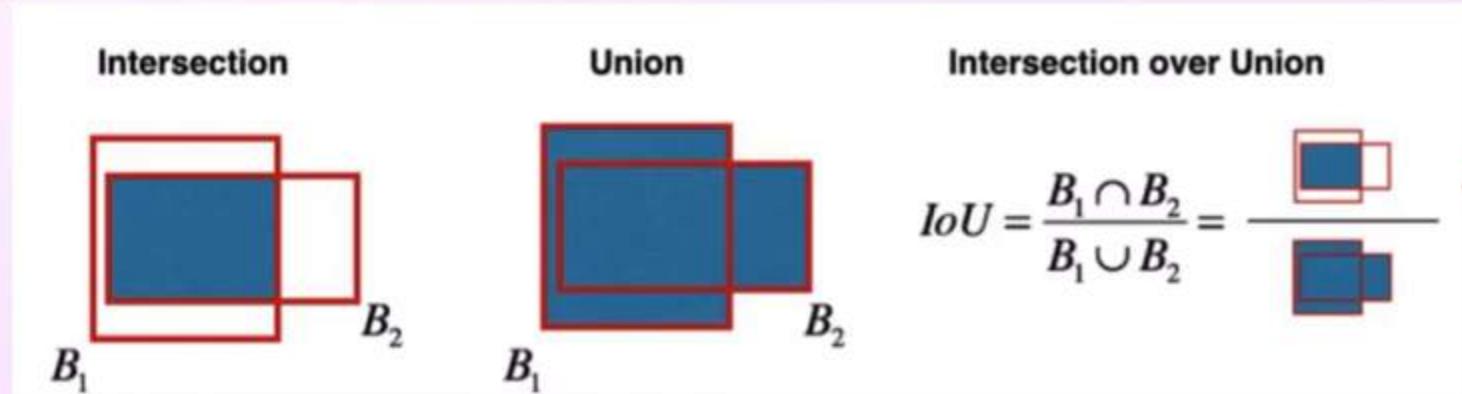
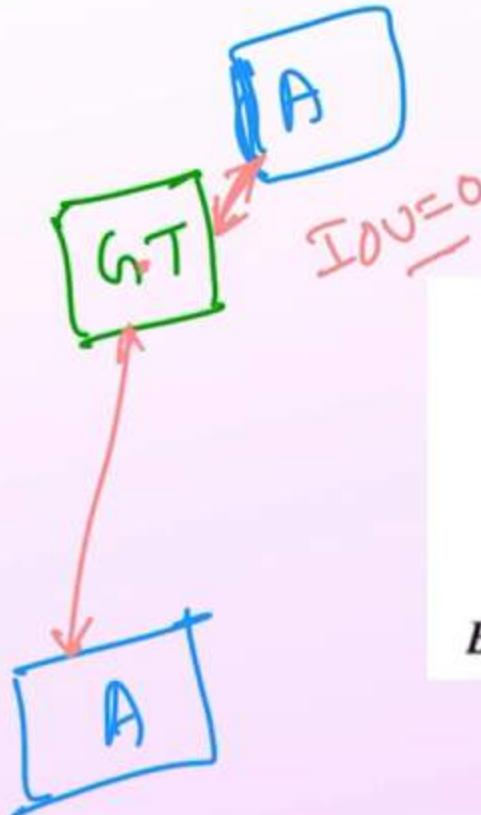
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$\overbrace{\hspace{10em}}^{\text{Loss} \downarrow}$

$\overbrace{\hspace{10em}}^{\text{IOU} \quad 0.7}$

$\overbrace{\hspace{10em}}^{\text{0.3} \downarrow \frac{\text{non overlapping region}}{0.}}$

# IOU Loss



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

Log ↓

IOU ↓

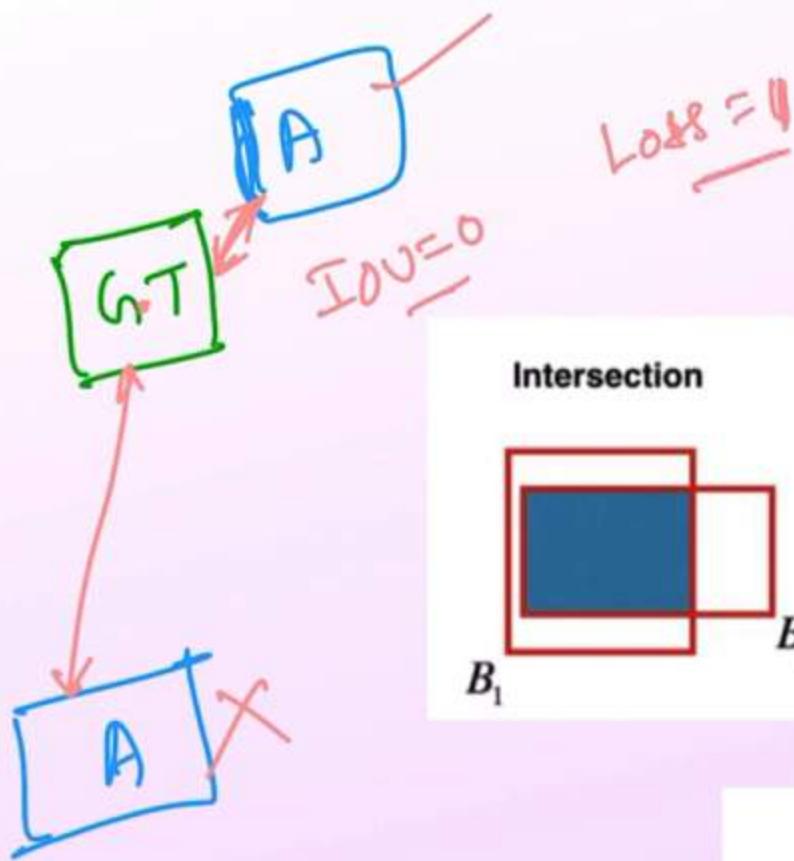
0.7

0.3 ↓ non overlapping region

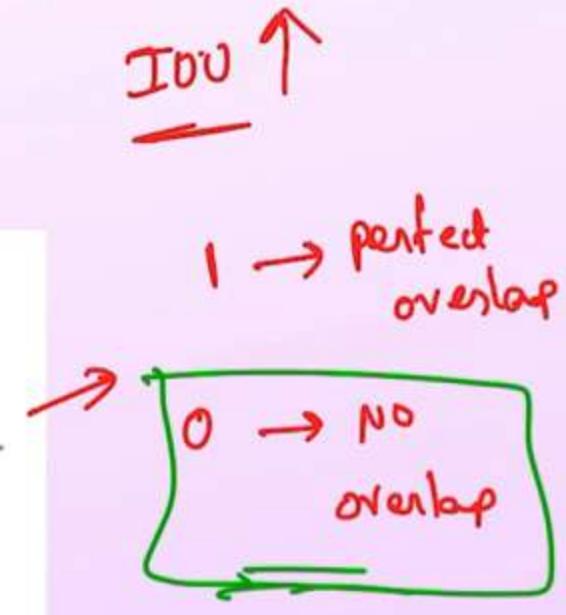
$\rightarrow 1 - IoU \rightarrow$  ↓

$\rightarrow$  ↓

# IOU Loss



$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2}$$



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

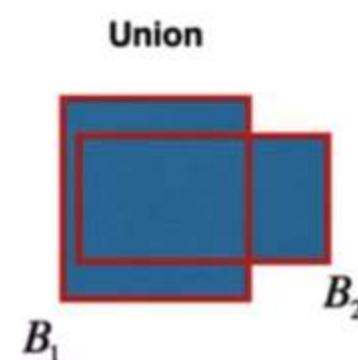
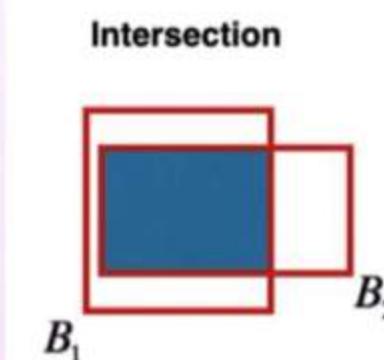
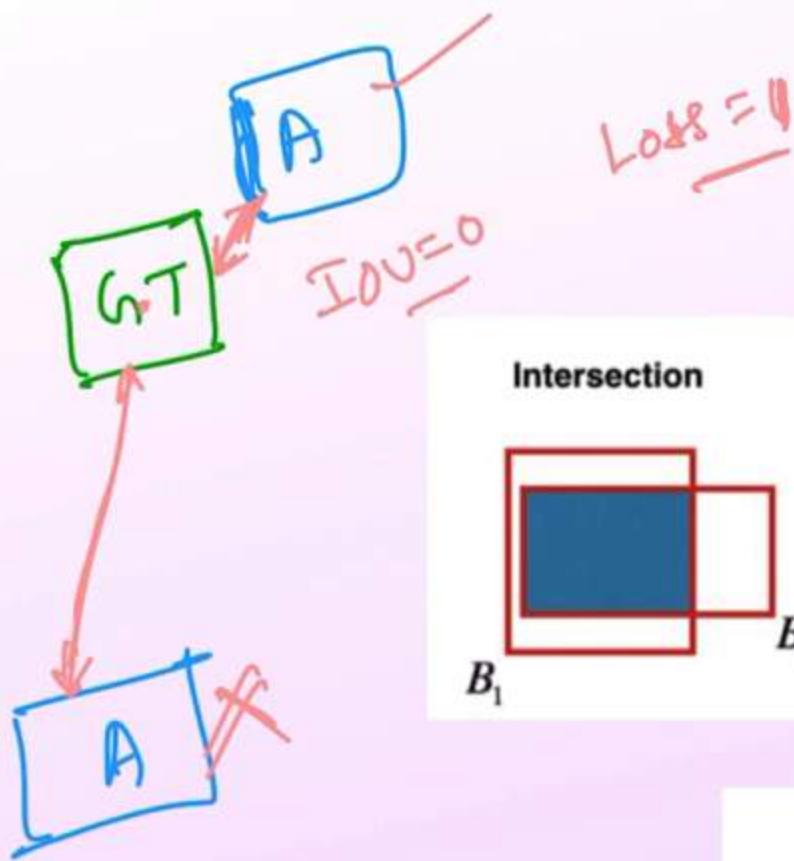
$\text{Log loss} \downarrow$

$\frac{\text{IOU}}{0.7}$

$\frac{0.3}{0.7}$

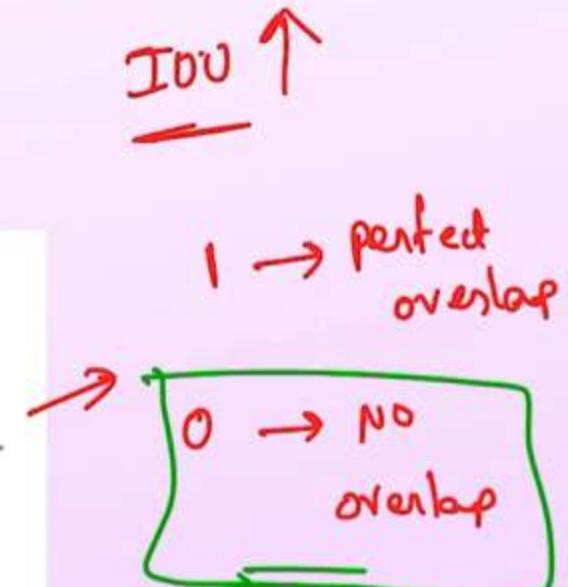
$\frac{\text{non overlapping region}}{0.7}$

# IOU Loss



**Intersection over Union**

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$



$$IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

$B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$  is the ground-truth

$B = (x, y, w, h)$  is the predicted box

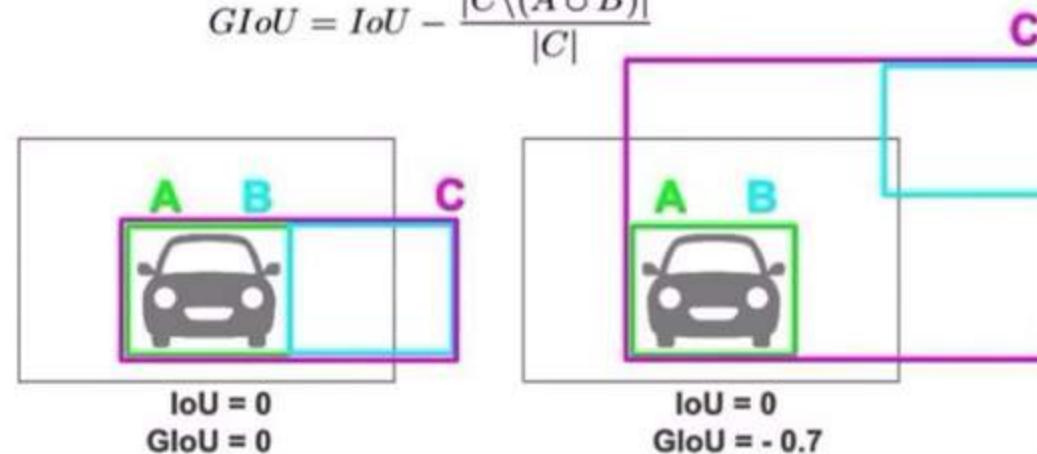
$$\mathcal{L}_{IoU} = 1 - \frac{B \cap B^{gt}}{B \cup B^{gt}}$$

**Loss ↓**

**IOU**  
0.7  
0.3  
0.0  
non overlapping region

# Generalized IOU Loss

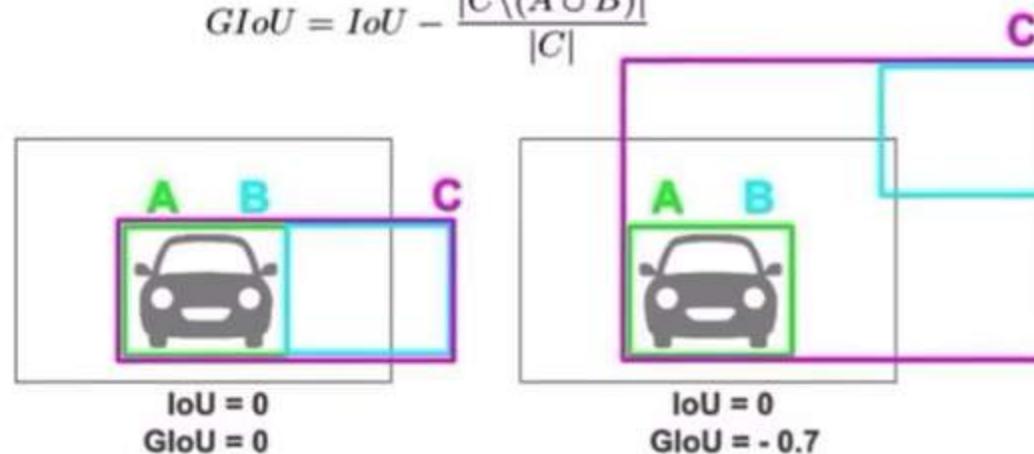
$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

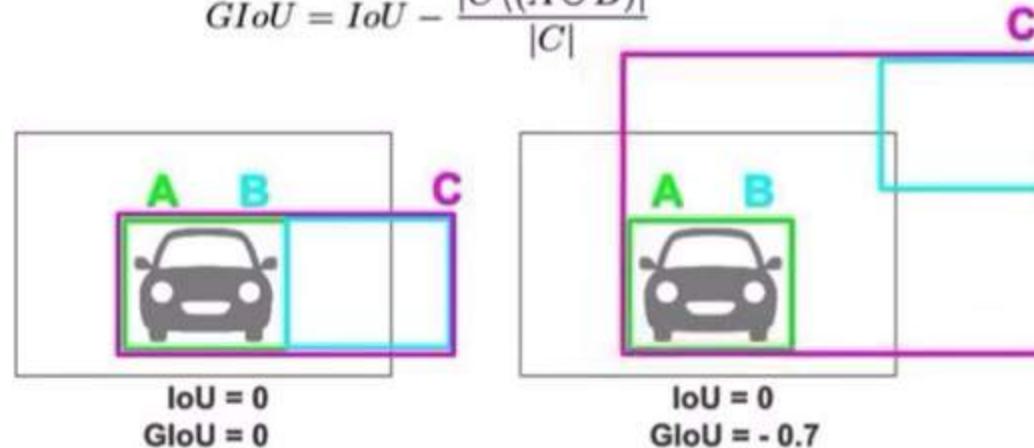
$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

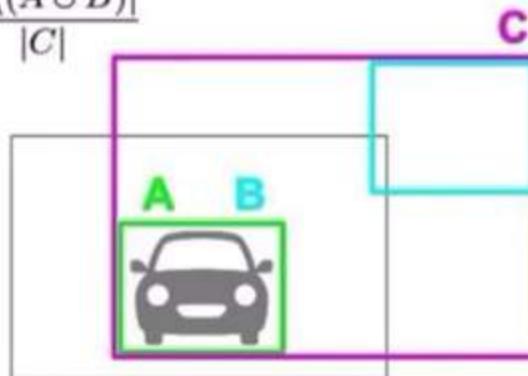
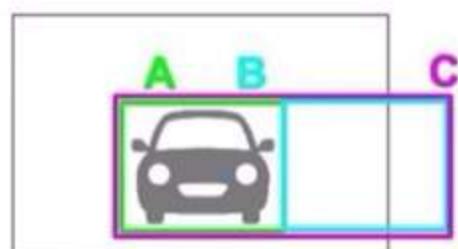
$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

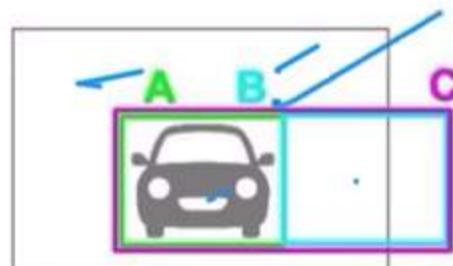


minimum  
Enclosed  
rectangle

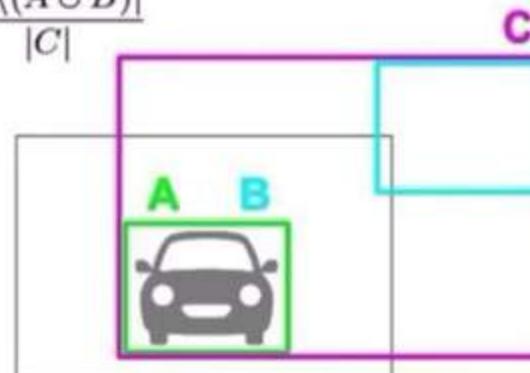
$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



IoU = 0  
GIoU = 0



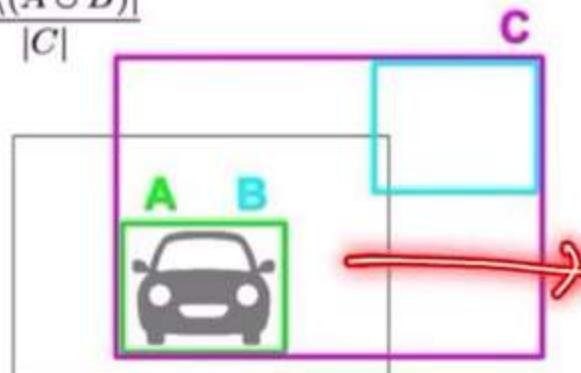
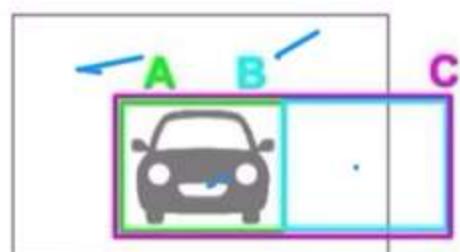
IoU = 0  
GIoU = -0.7

minimum  
Enclosed  
rectangle

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

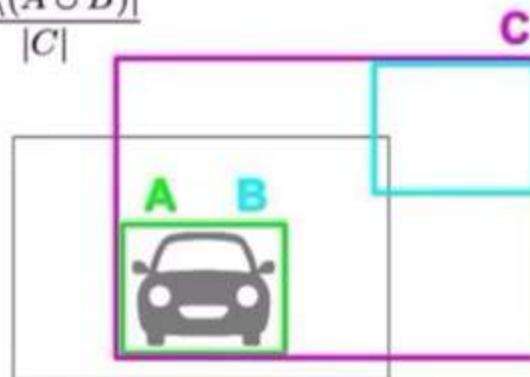
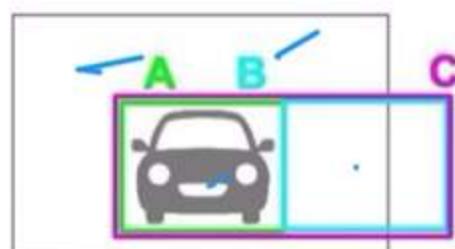


minimum  
Enclosed  
rectangle

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

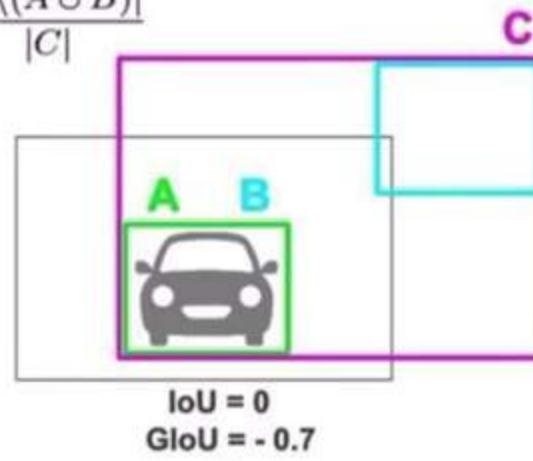
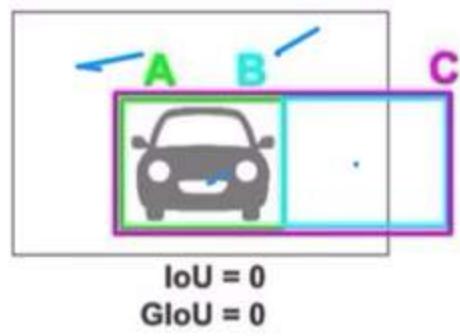


minimum  
Enclosed  
rectangle  
Area

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

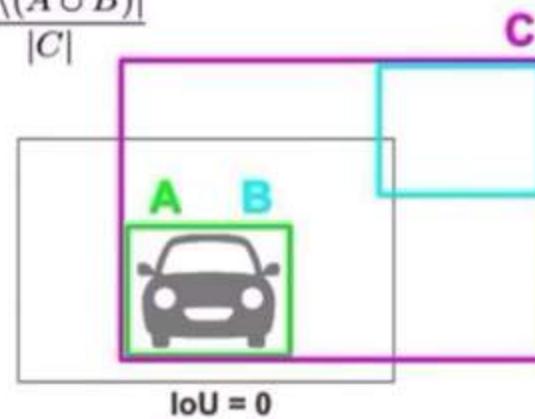
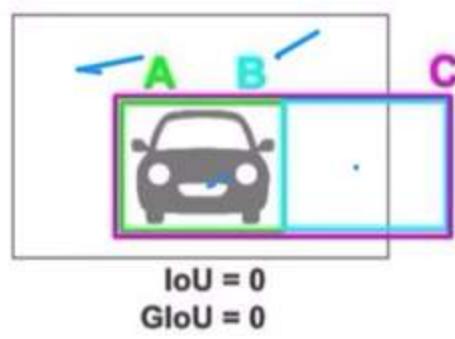


minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$

$$\mathcal{L}_{GIoU} = 1 - \underline{IoU} + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

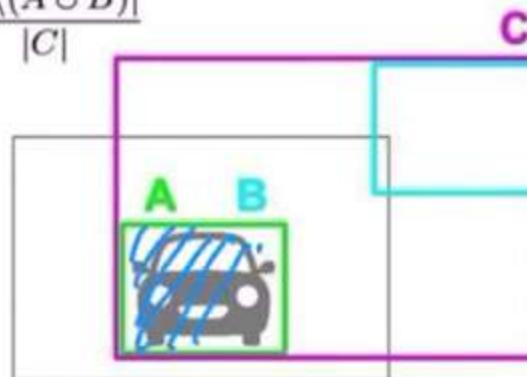
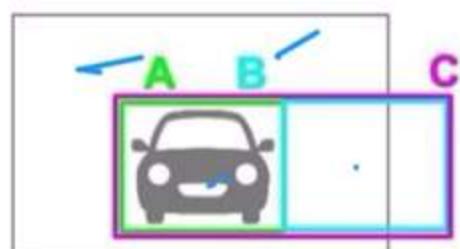


minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$   
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

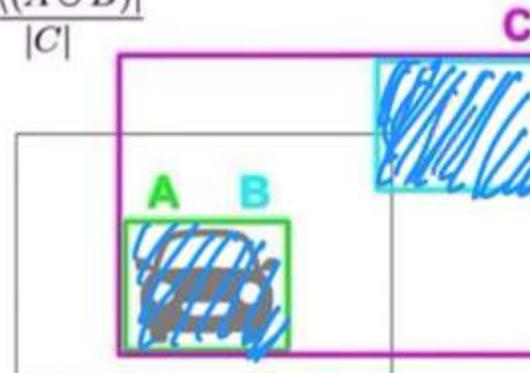
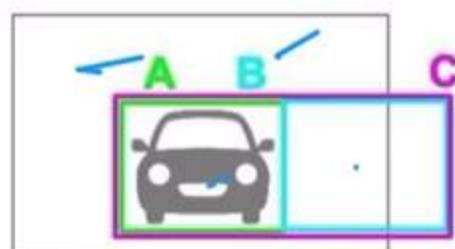


minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$   
↓  
down

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

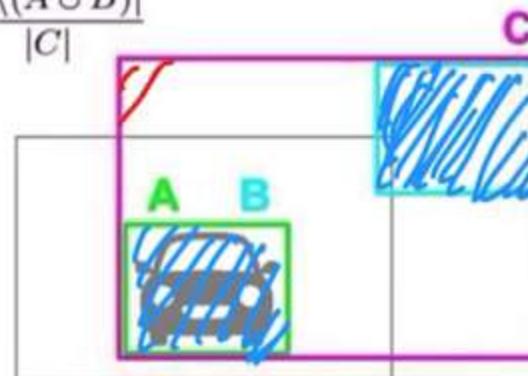
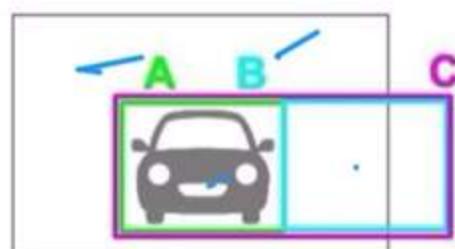


minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$   
↓  
down

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$

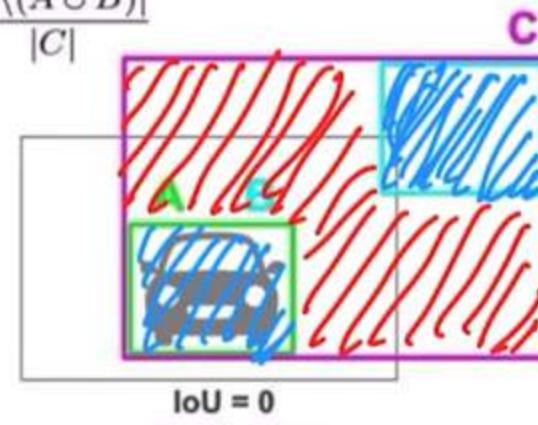
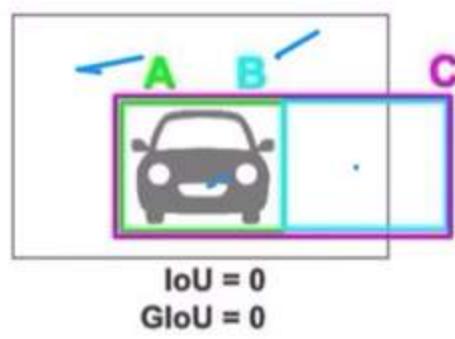


minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$   
↓  
down

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



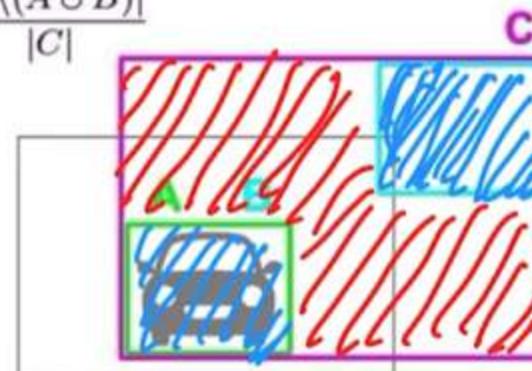
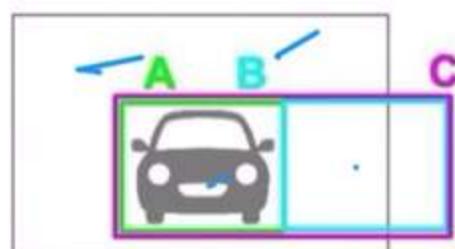
minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$

↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow 0, 1$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



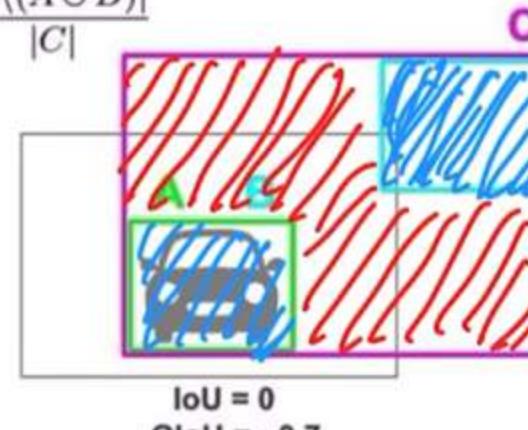
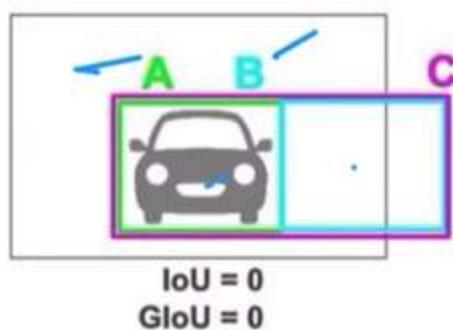
minimum  
Enclosed  
Rectangle  
 $\text{Area}(C) > \text{for}$

↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow (0, 1)$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



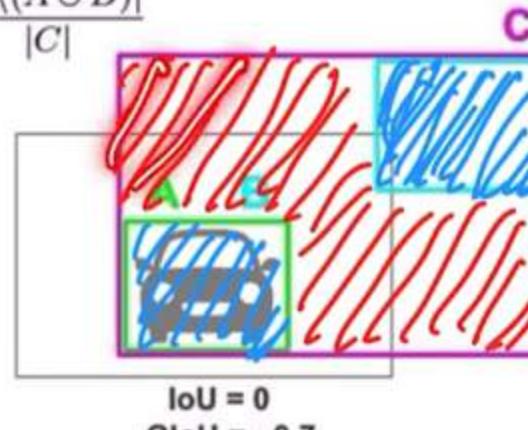
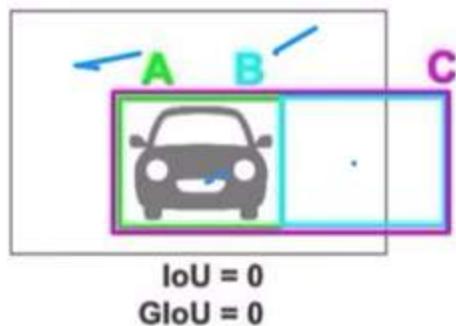
minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$   
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - 80%  
↑

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$

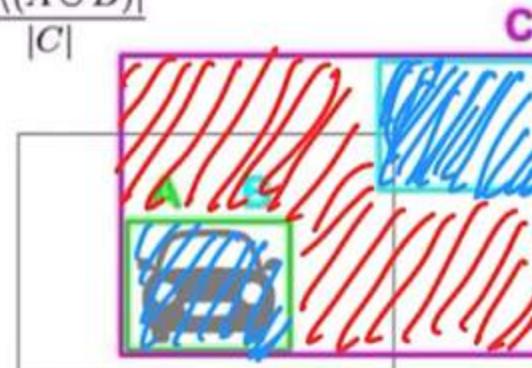
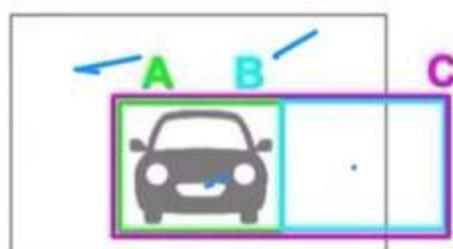
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1] \downarrow$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$

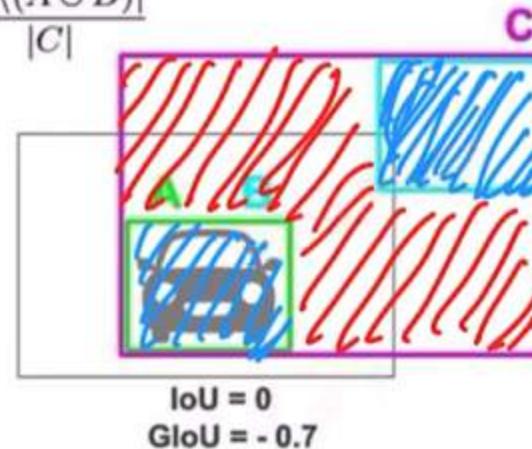
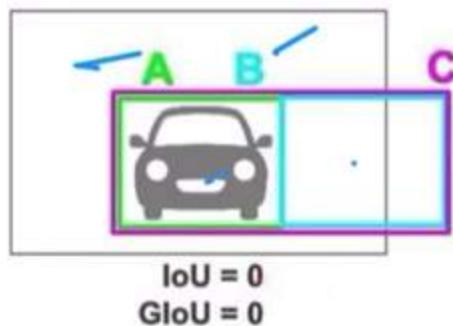
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{far}$

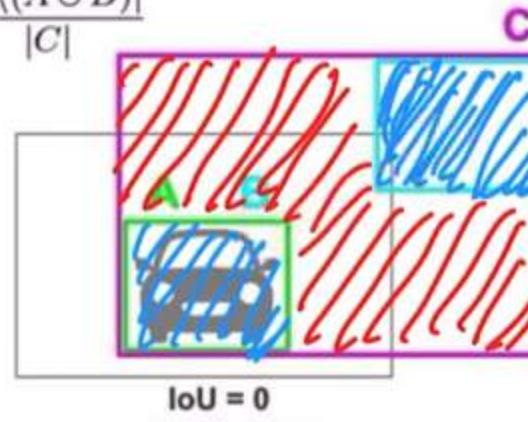
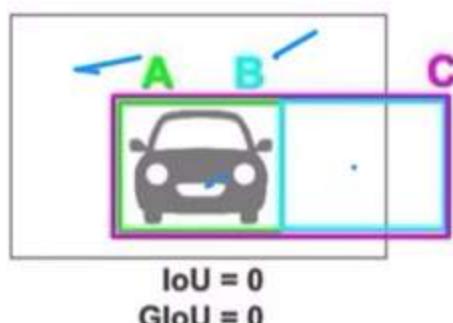
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(c) > \text{far}$

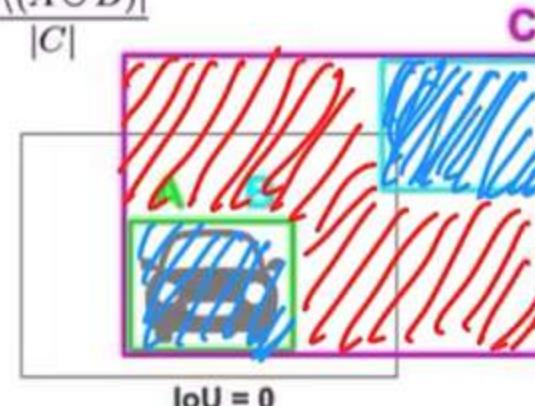
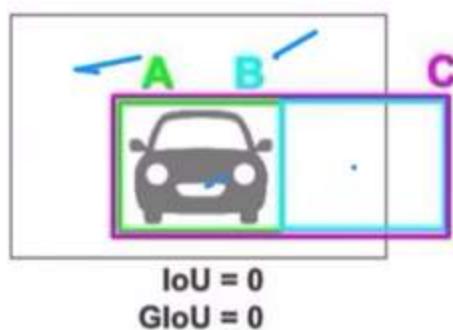
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle

$\text{Area}(C) > \text{for}$

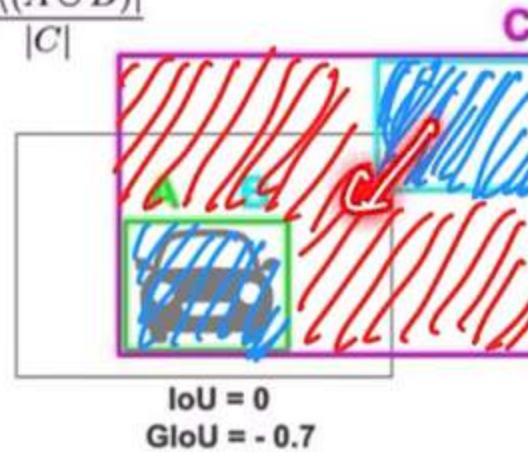
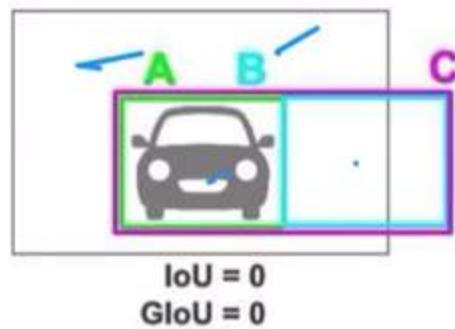
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{far}$

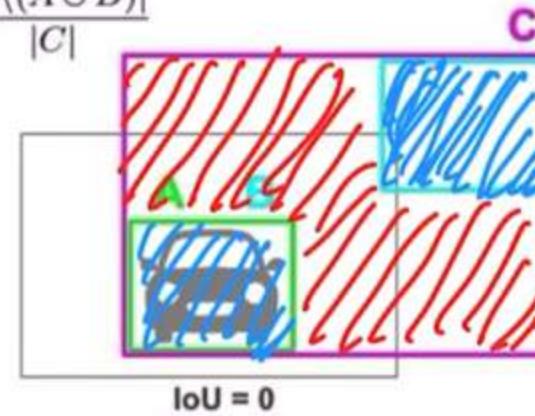
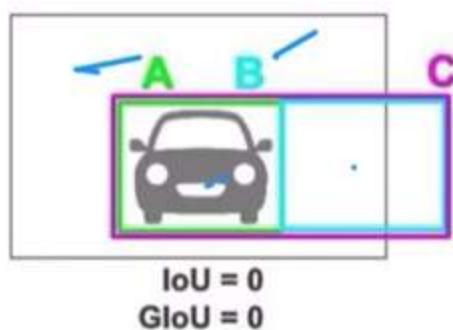
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle

$\text{Area}(C) > \text{for}$

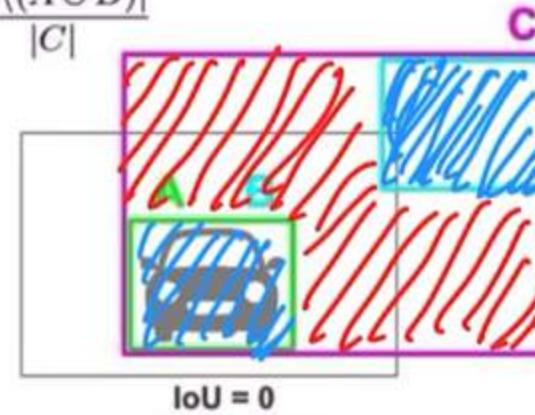
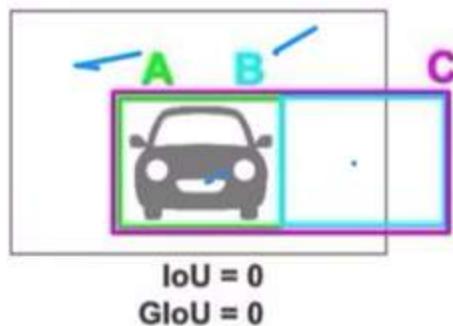
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1] \downarrow$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle

$\text{Area}(C) > \text{for}$

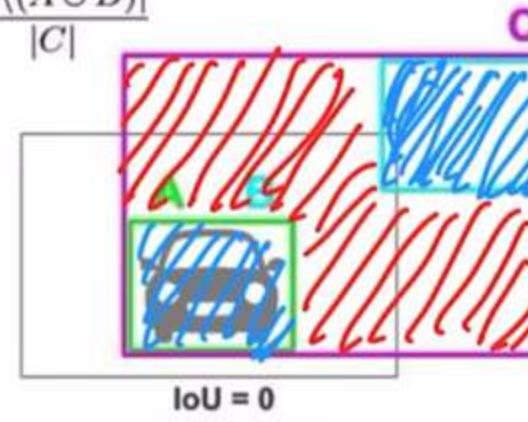
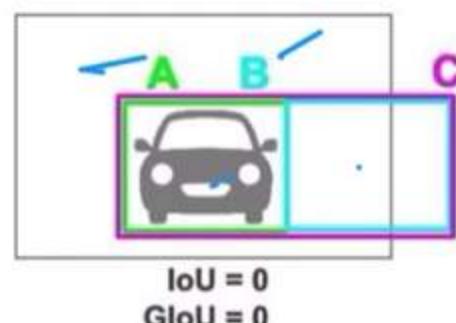
↓  
down

C -  $\overline{\text{B} \cup \text{B}^{gt}}$

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle

$\text{Area}(C) > \text{for}$

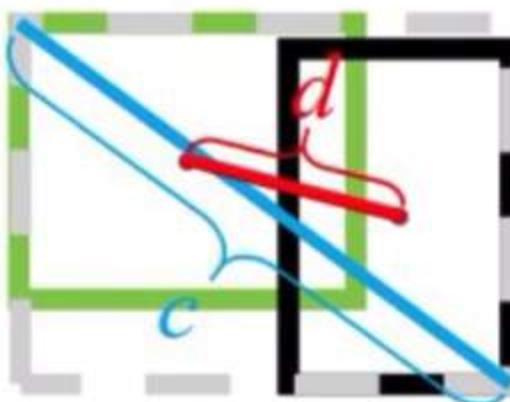
↓  
down

C -  $\overline{\text{B} \cup \text{B}^{gt}}$

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$



$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

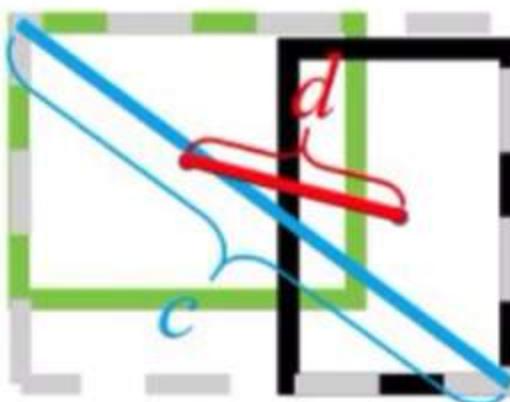
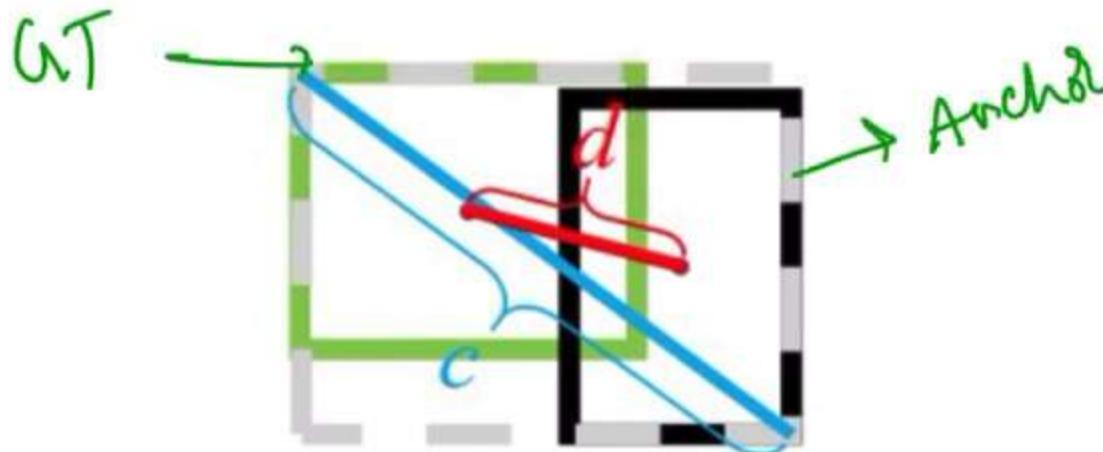


Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

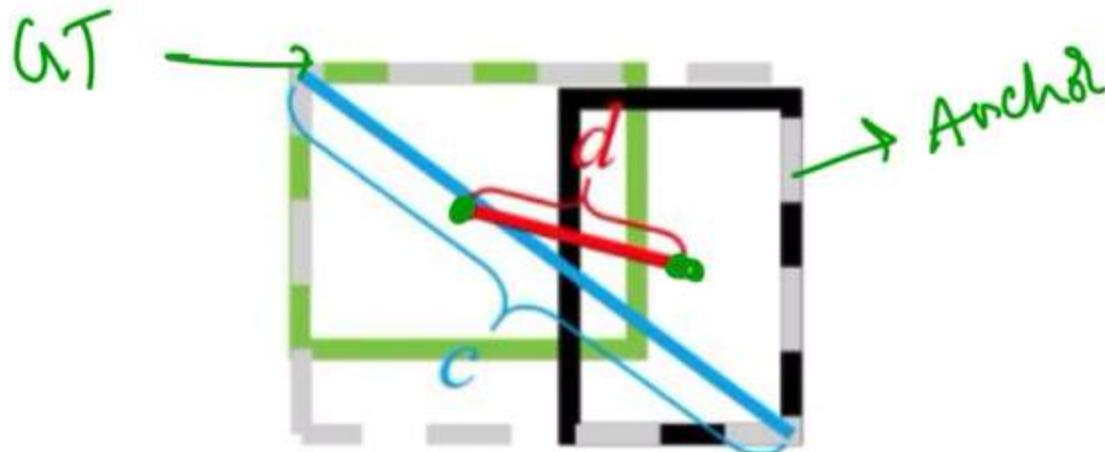


$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

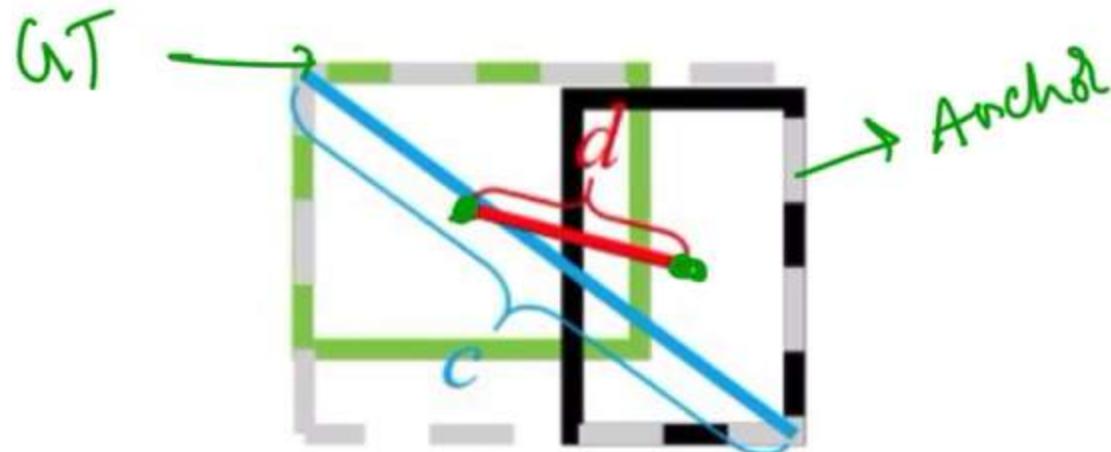


$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

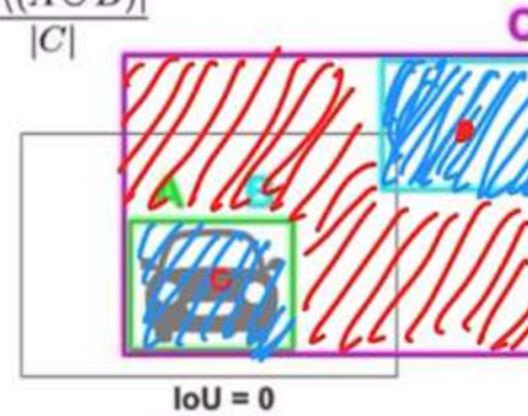
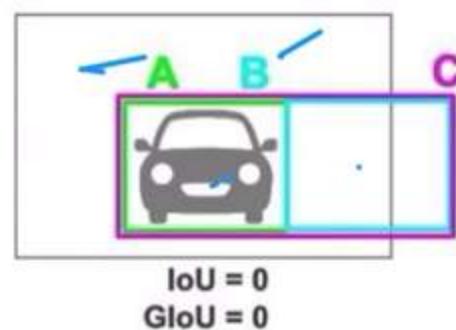


$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle  
 $\text{Area}(C) > \text{for}$

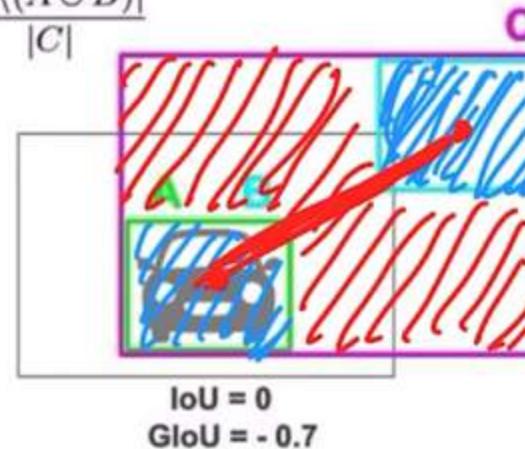
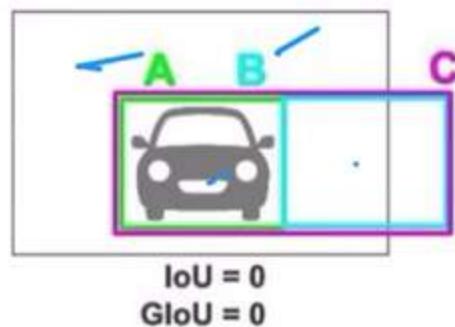
↓  
low

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

C - B ∪ B<sup>gt</sup>

# Generalized IOU Loss

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}$$



minimum  
Enclosed  
rectangle

$\text{Area}(C) > \text{for}$

↓  
down

C - B ∪ B<sup>gt</sup>

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|} \rightarrow [0, 1]$$

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$



$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

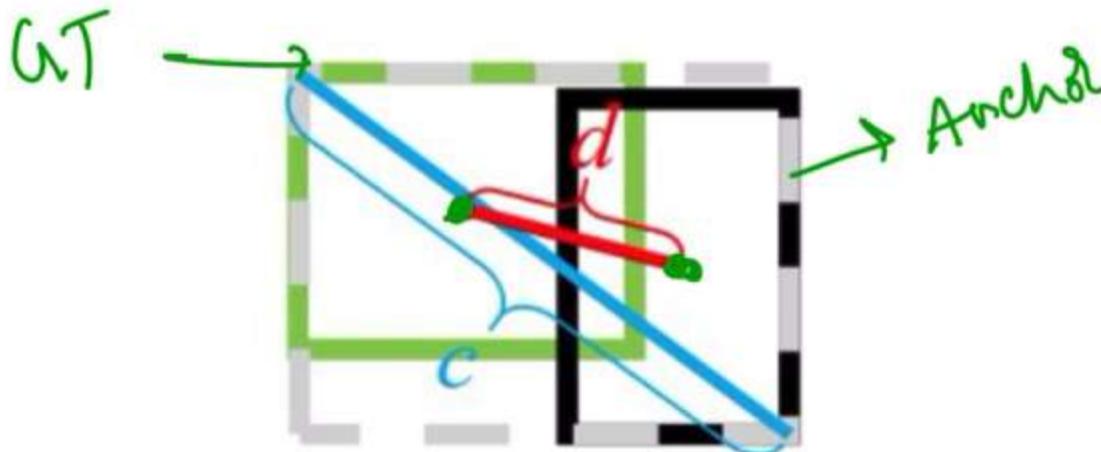


Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

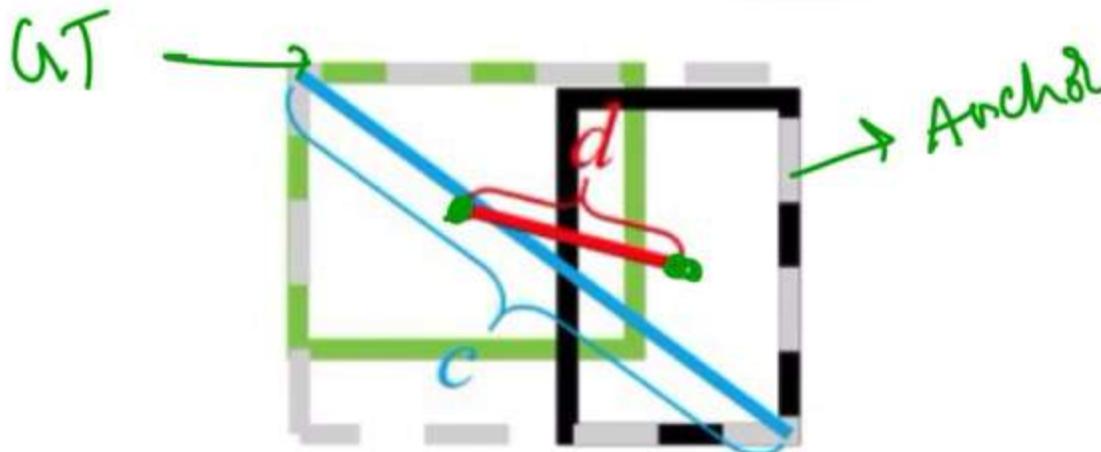


$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

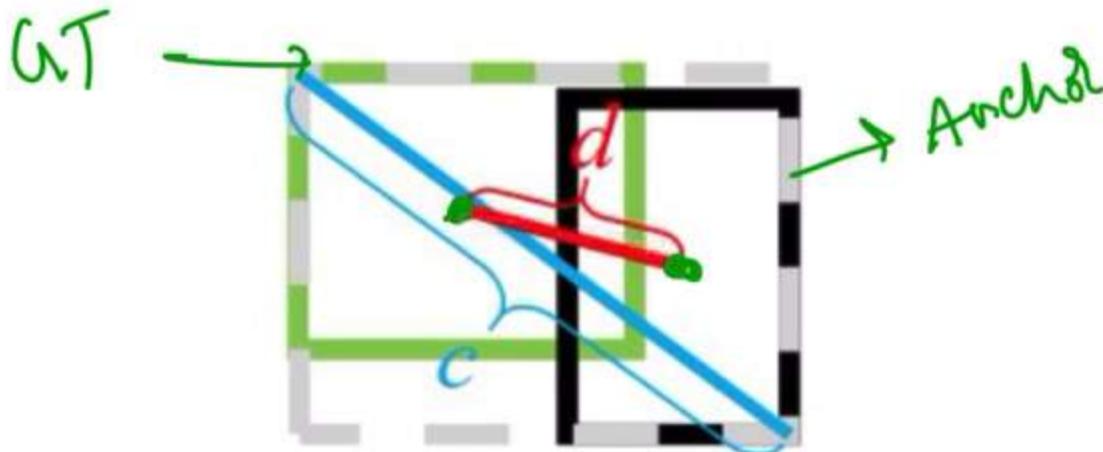


$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^g|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

# Distance IOU Loss

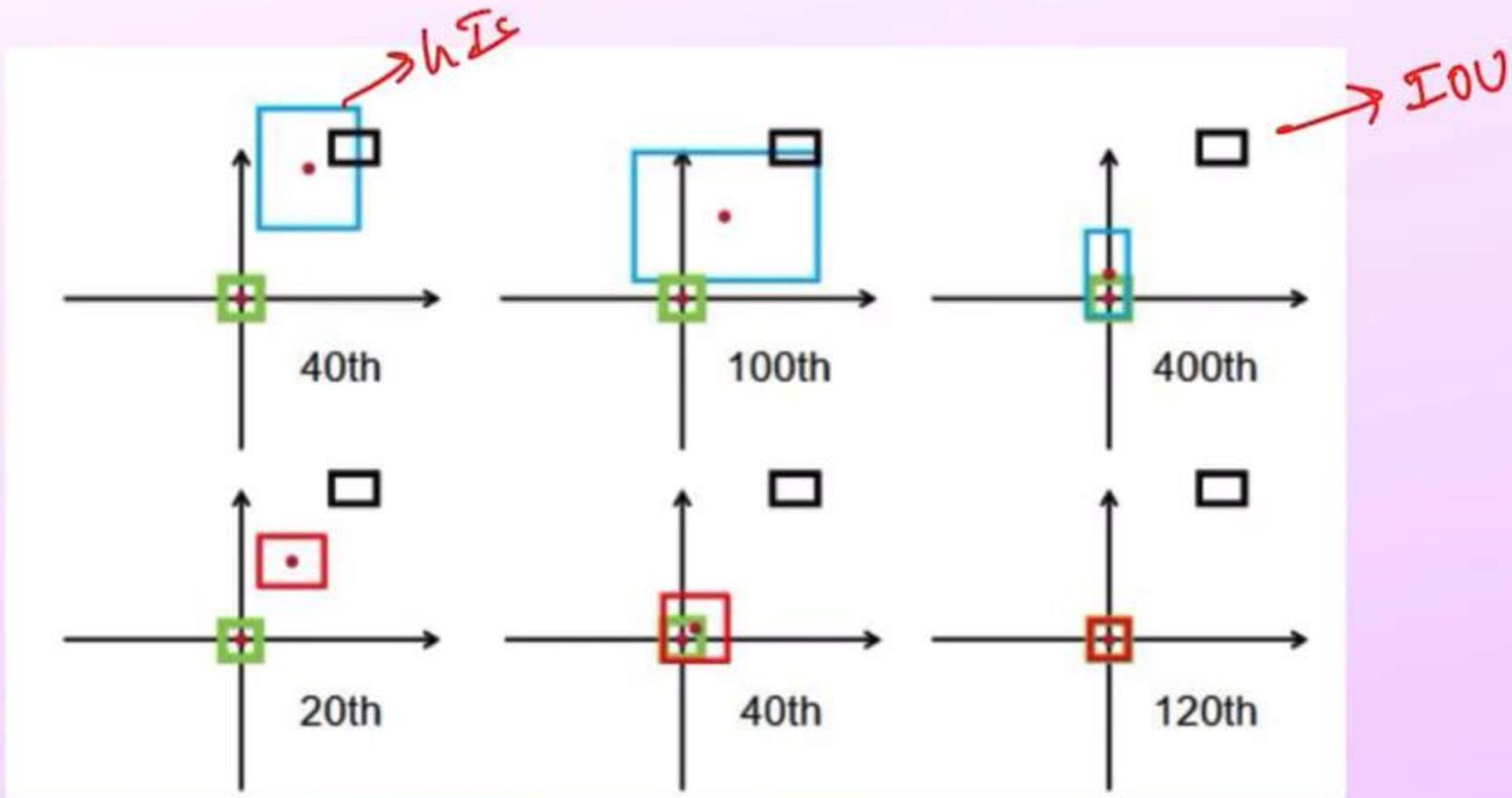
$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$



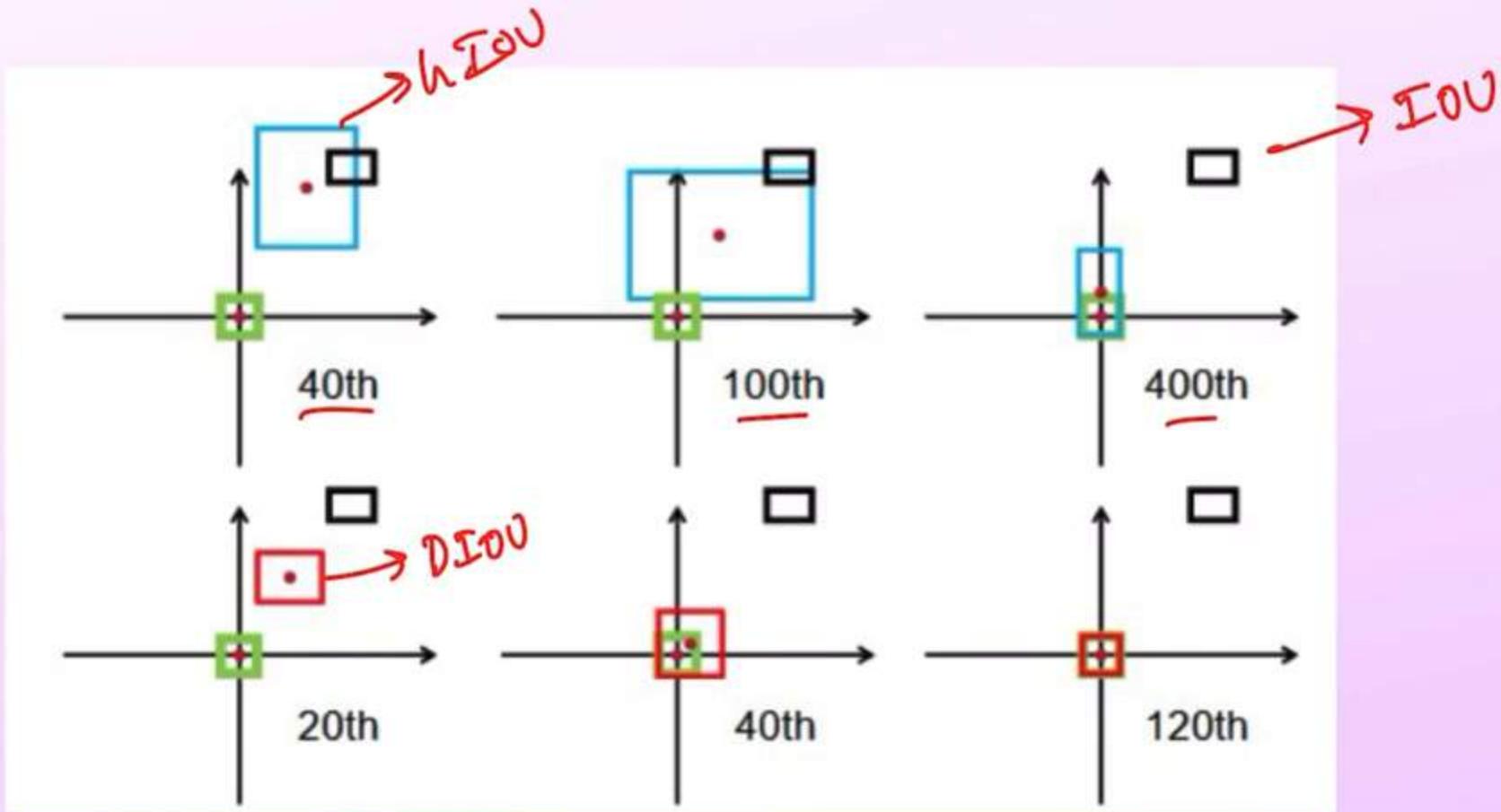
$$\mathcal{L}_{GIoU} = 1 - IoU + \frac{|C - B \cup B^{gt}|}{|C|}$$

Figure 5: DIoU loss for bounding box regression, where the normalized distance between central points can be directly minimized.  $c$  is the diagonal length of the smallest enclosing box covering two boxes, and  $d = \rho(\mathbf{b}, \mathbf{b}^{gt})$  is the distance of central points of two boxes.

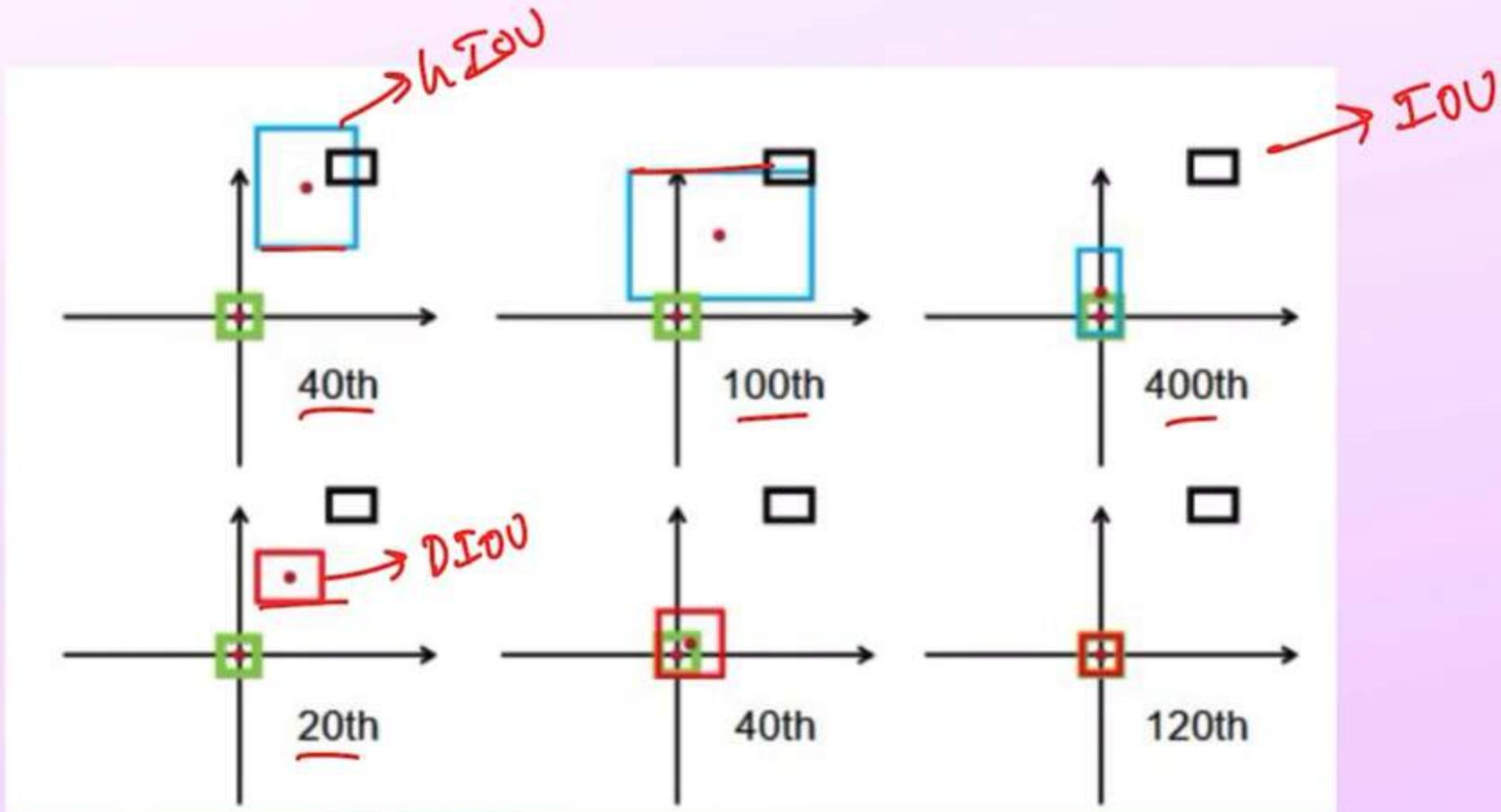
# Distance IOU Loss



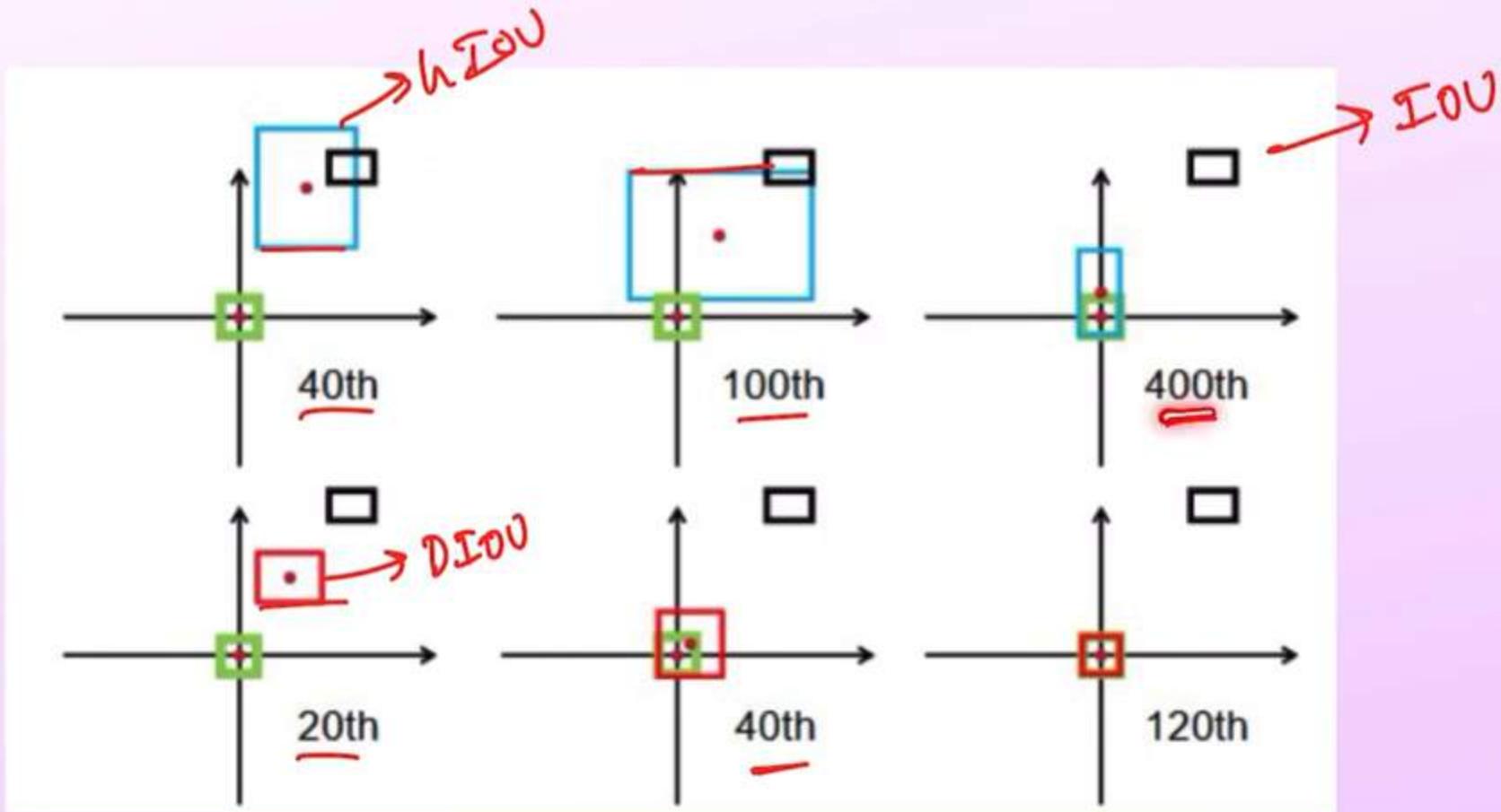
# Distance IOU Loss



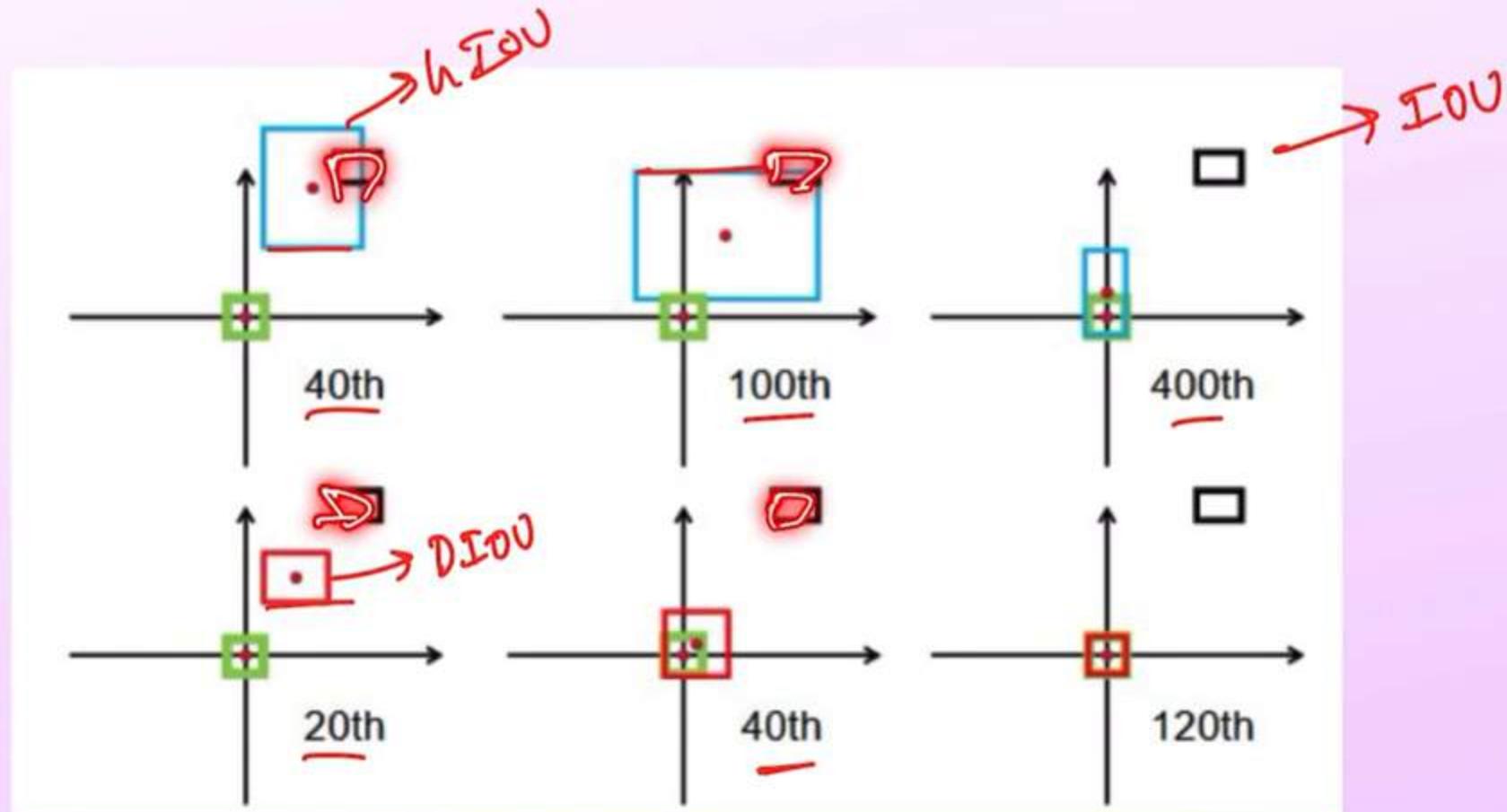
# Distance IOU Loss



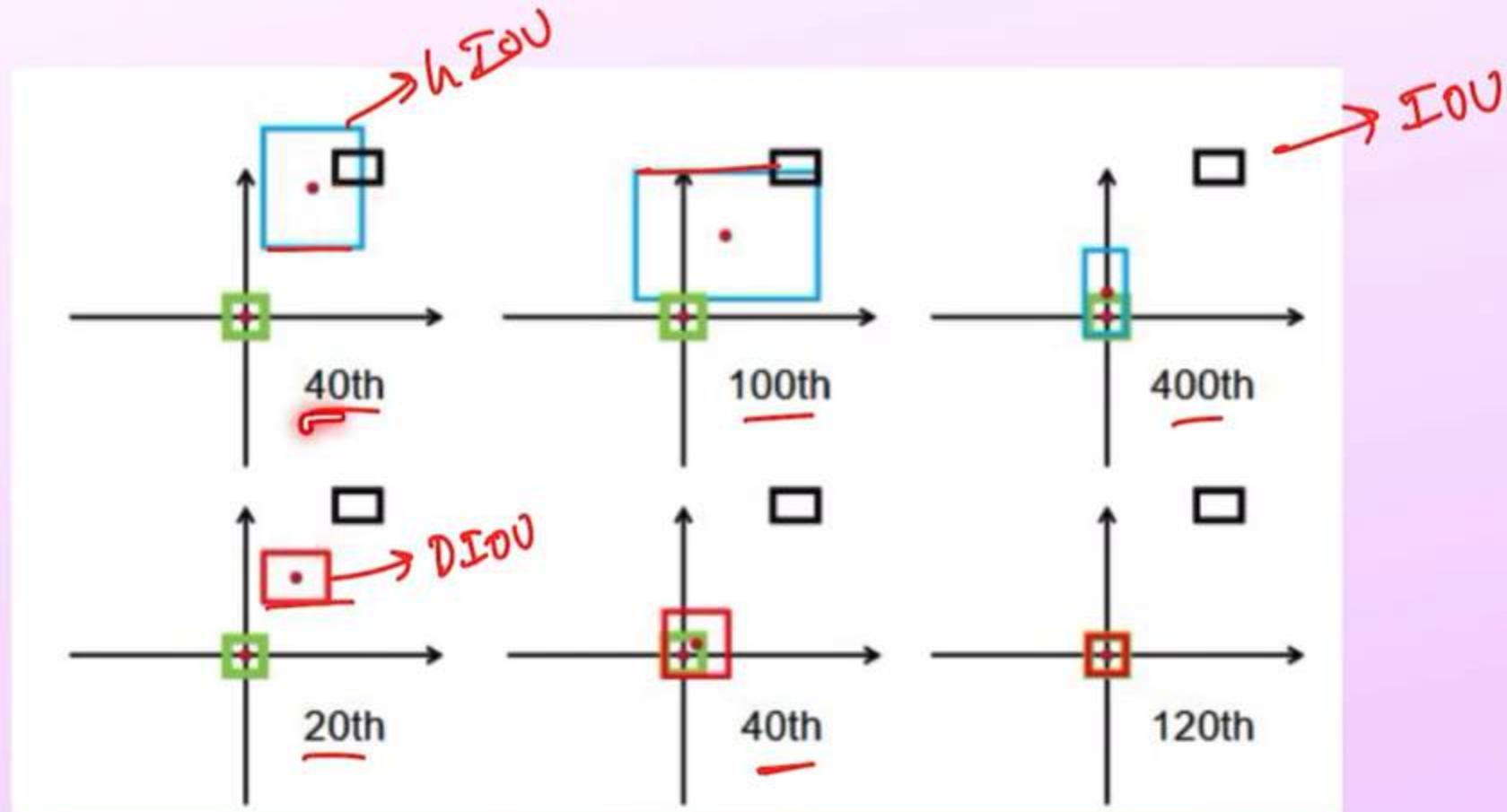
# Distance IOU Loss



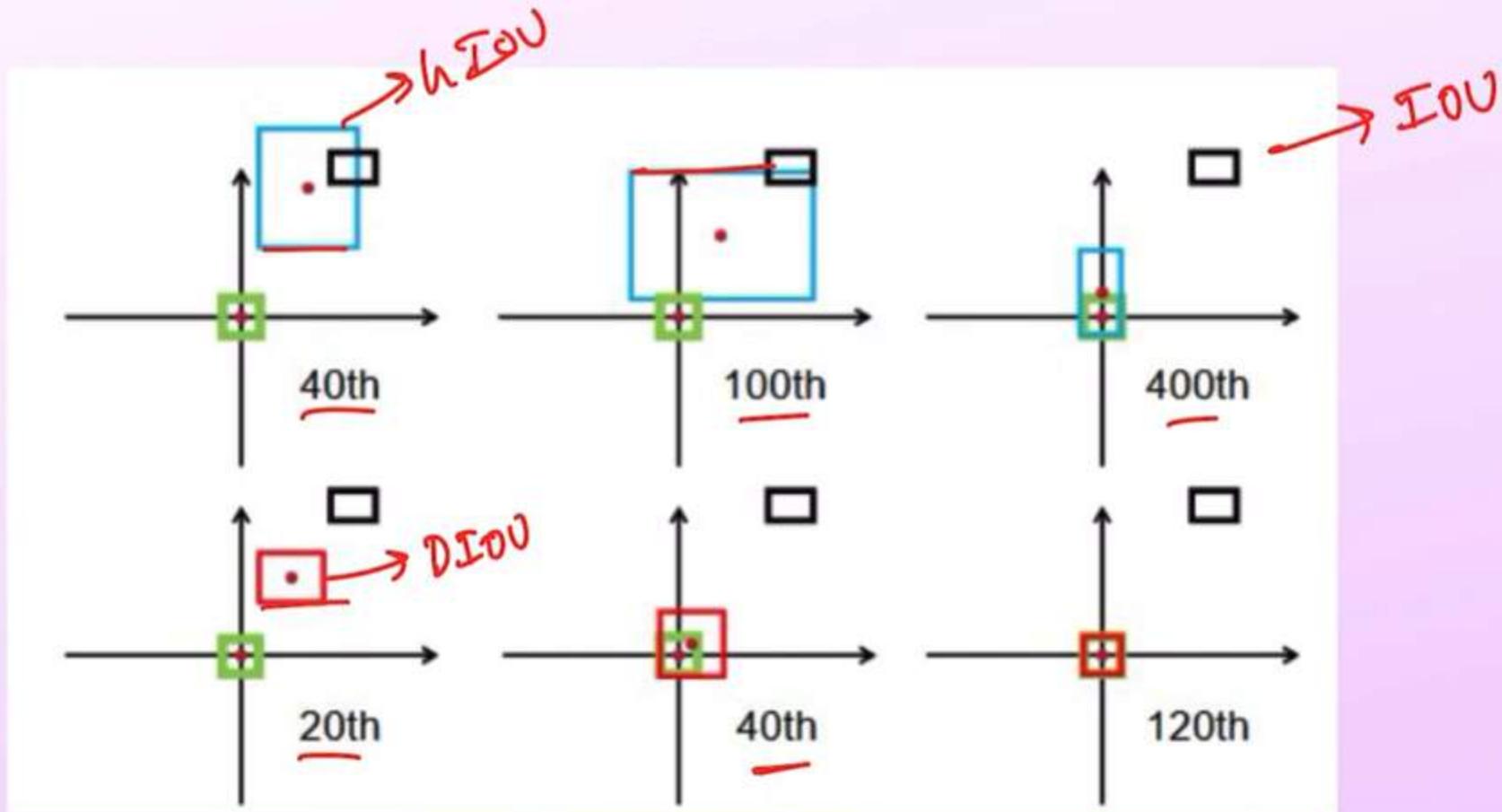
# Distance IOU Loss



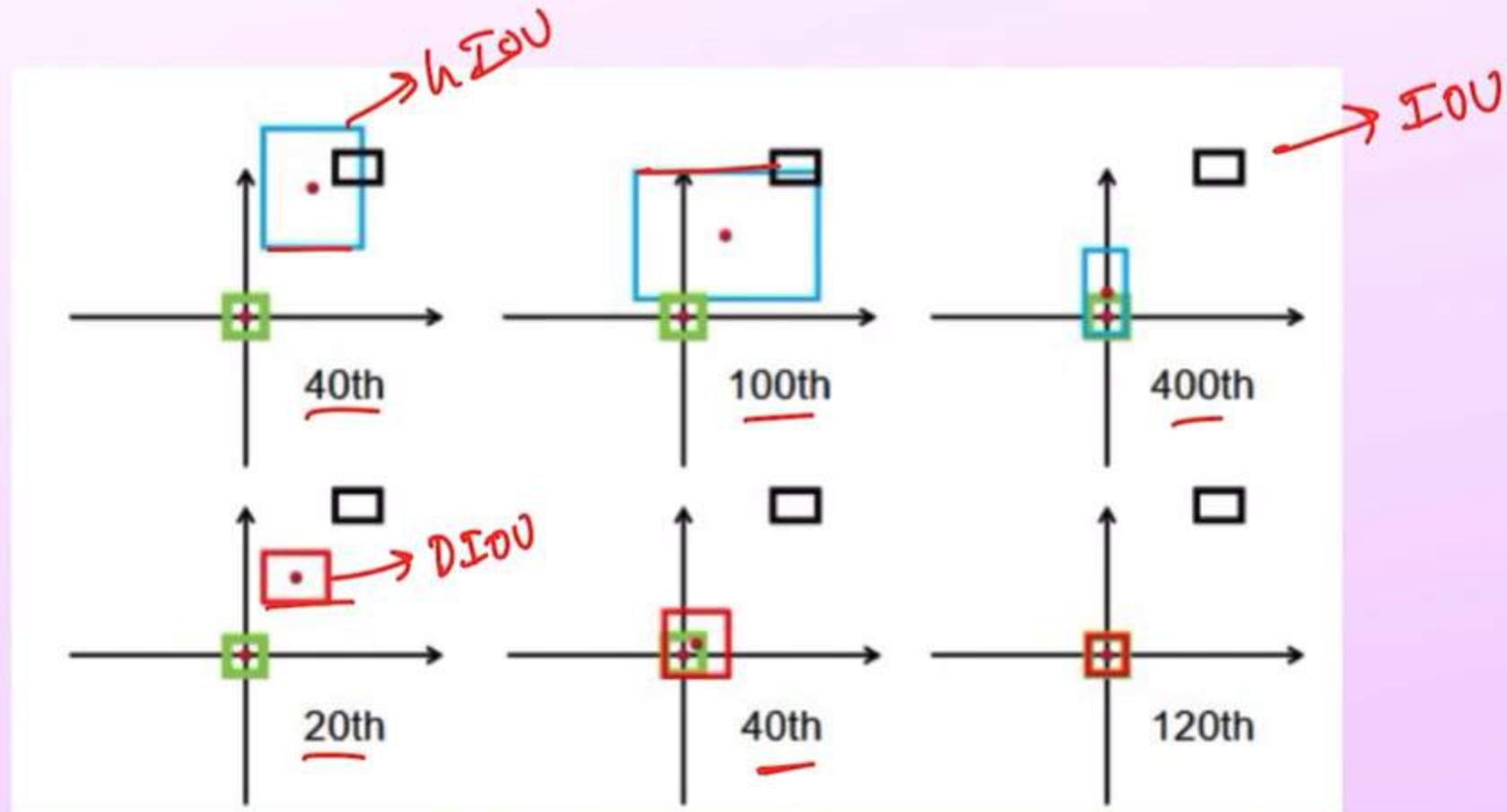
# Distance IOU Loss



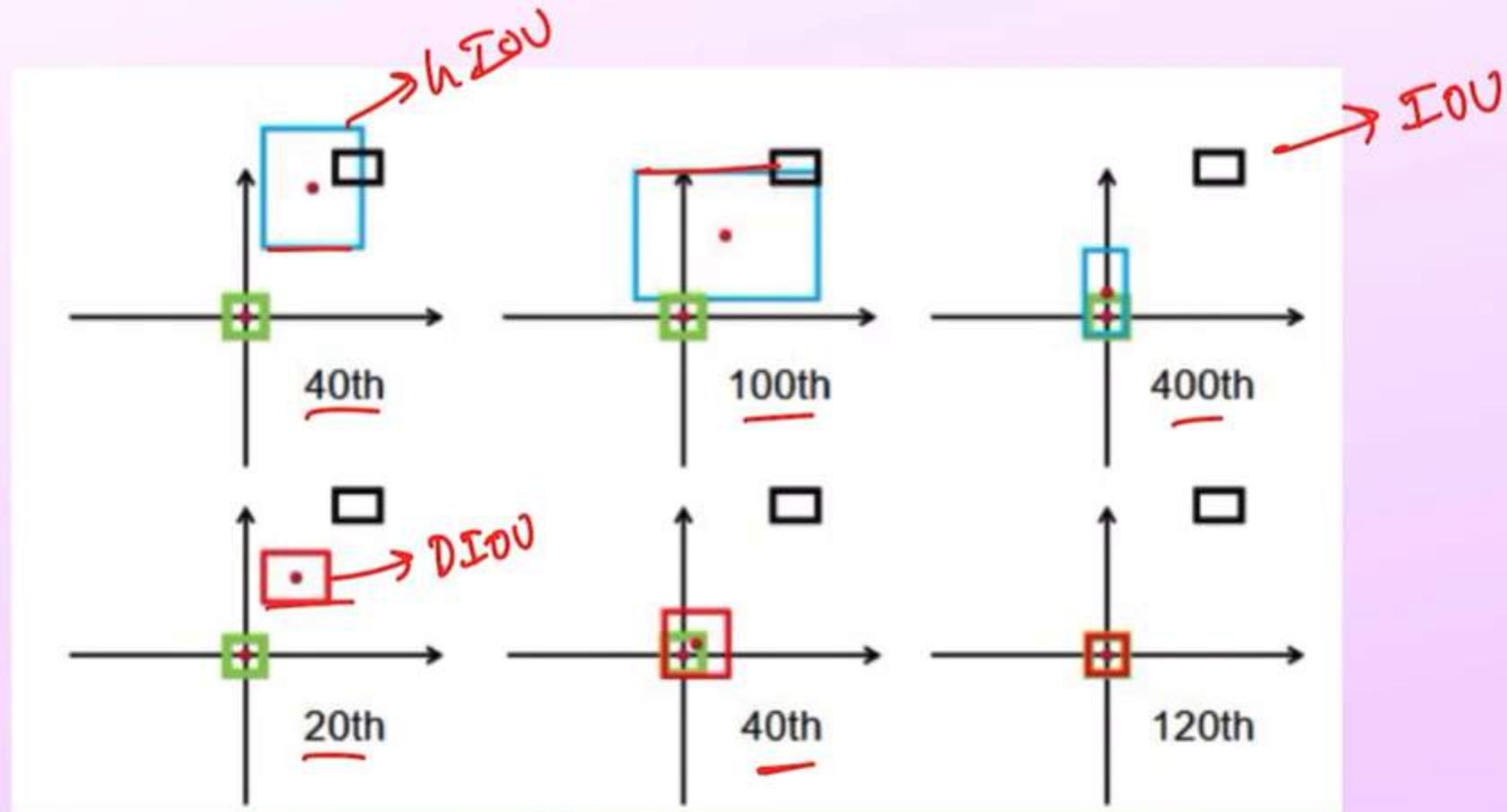
# Distance IOU Loss



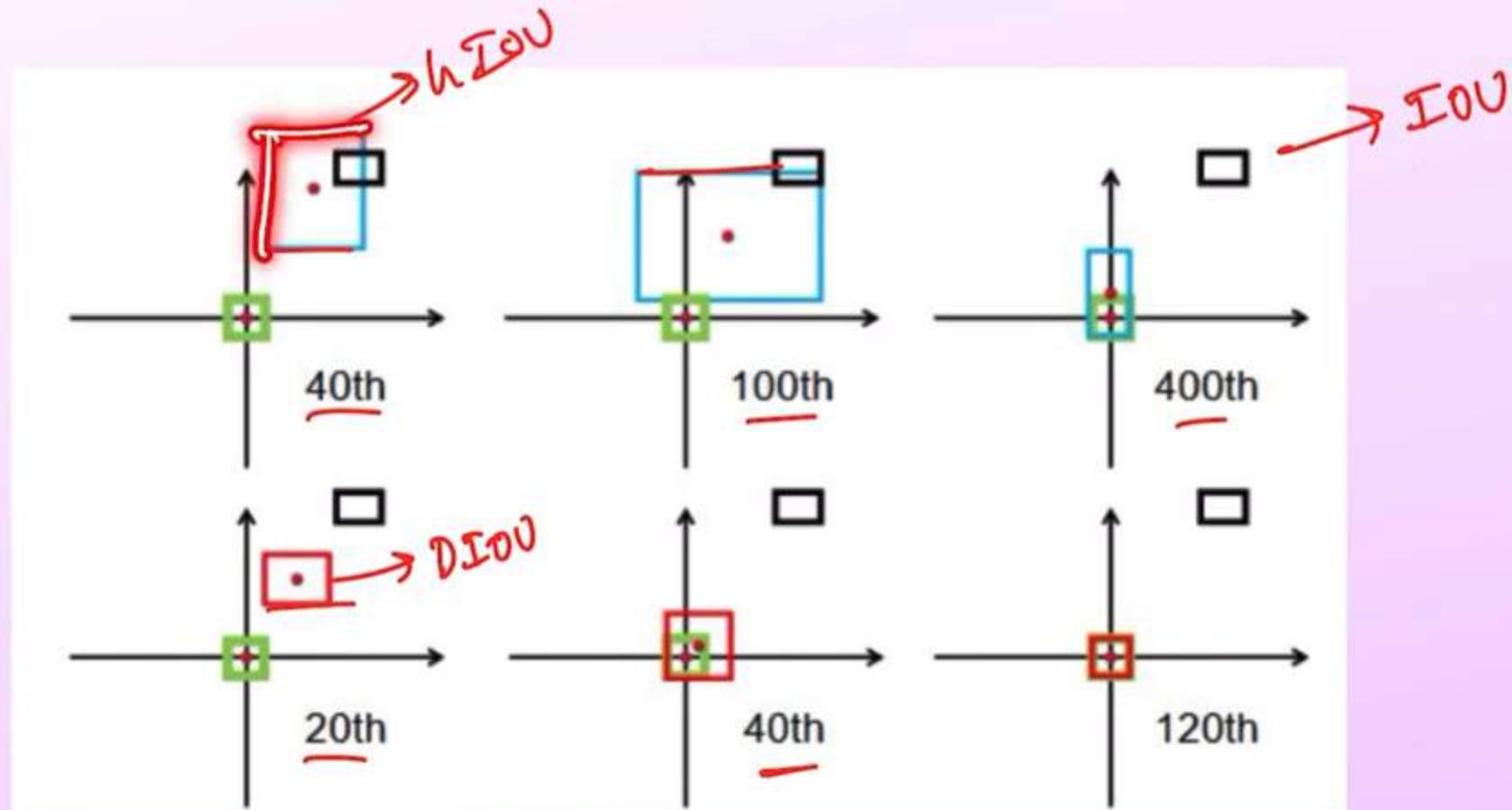
# Distance IOU Loss



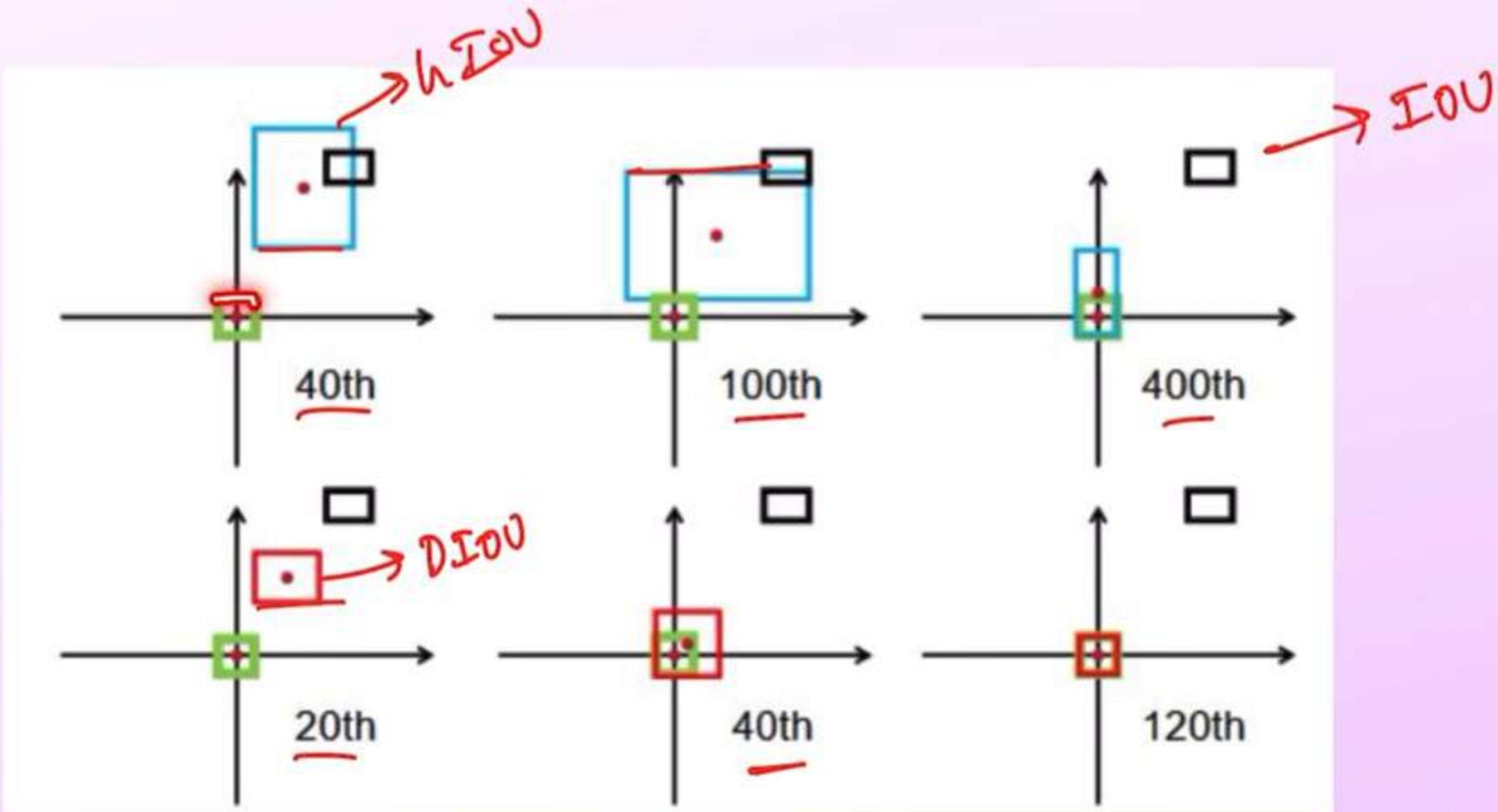
# Distance IOU Loss



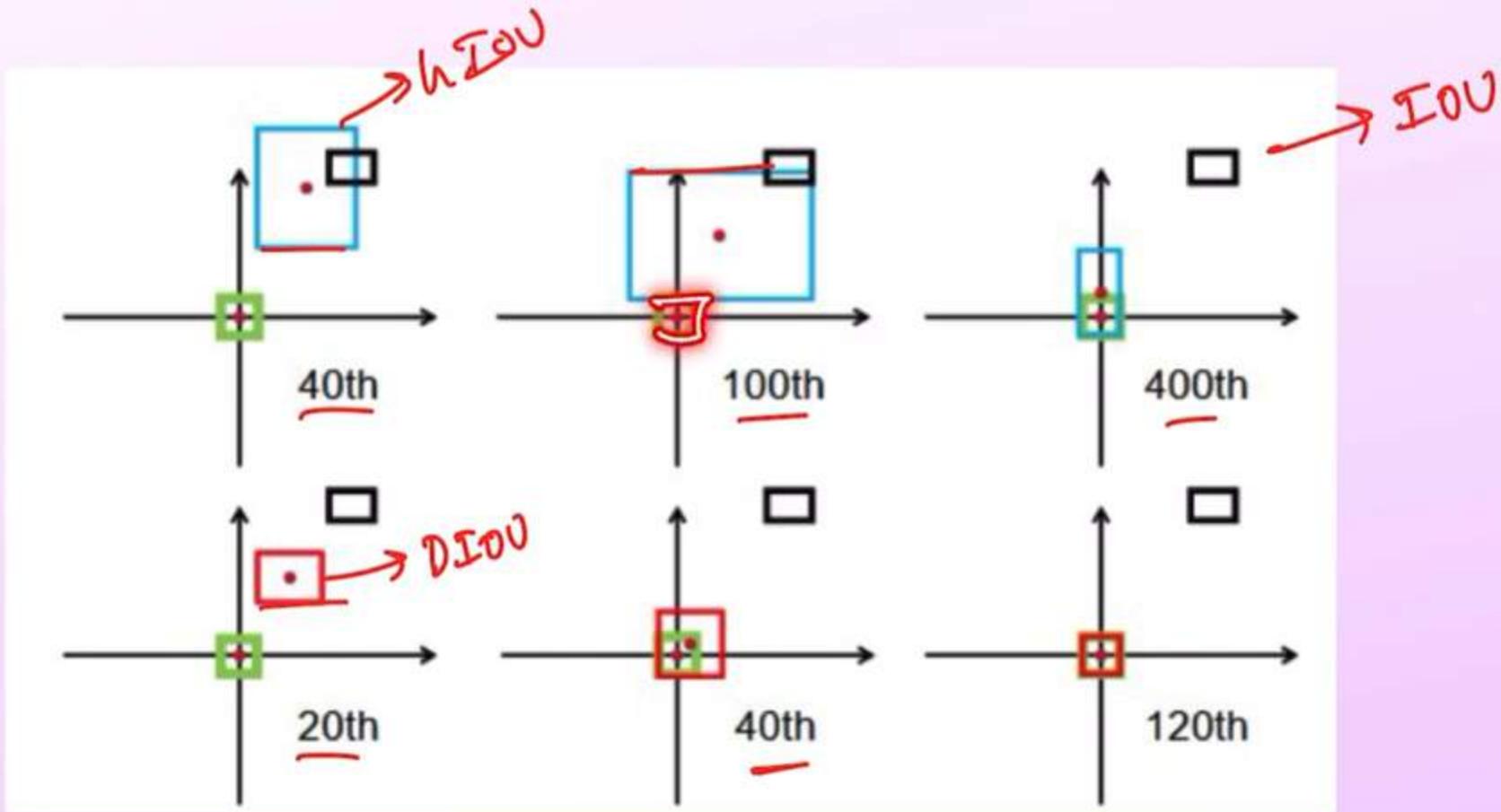
# Distance IOU Loss



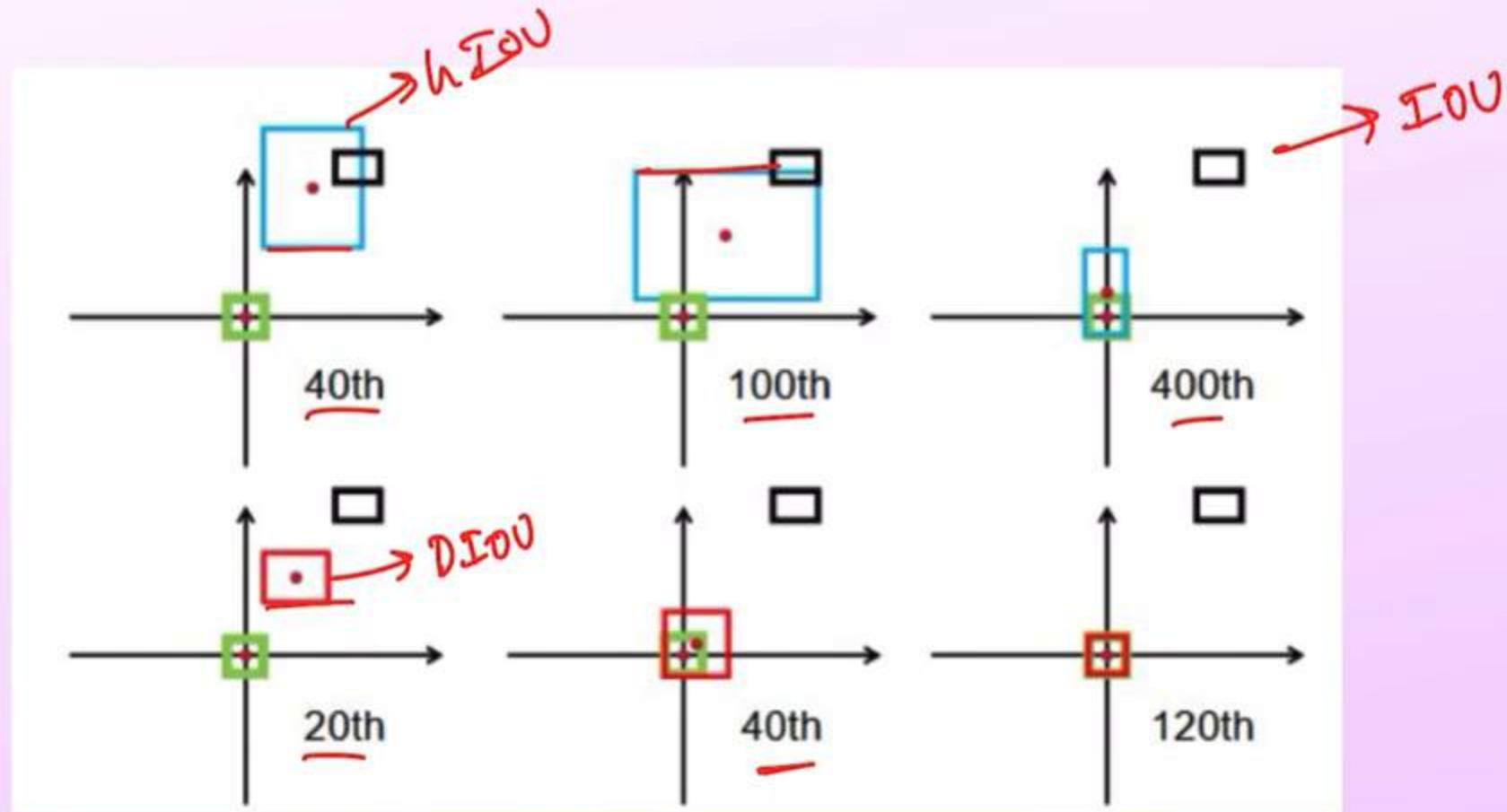
# Distance IOU Loss



# Distance IOU Loss



# Distance IOU Loss



# Complete IOU Loss

- . Overlapping Area
- . Distance between centers
- . Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

*Distance penalty*

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

*Aspect ratio*

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

With weight

$$v = \frac{4}{\pi^2} \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio.

with weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

With weight

$$v = \frac{4}{\pi^2} \left( arctan \frac{w^{gt}}{h^{gt}} - arctan \frac{w}{h} \right)^2$$
$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

With weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

With weight

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} \rightarrow \text{Distance penalty}$$

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \rightarrow \text{Aspect ratio.}$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

with weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

with weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

With weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

Distance penalty

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

Aspect ratio

With weight

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2$$

$$\alpha = \frac{v}{(1 - IoU) + v}$$

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

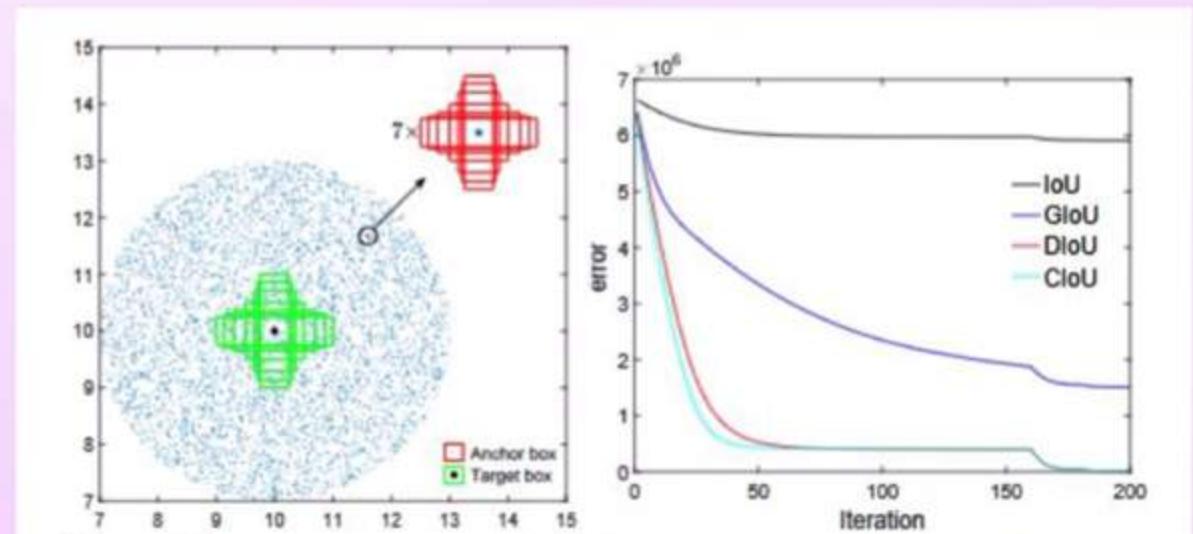


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

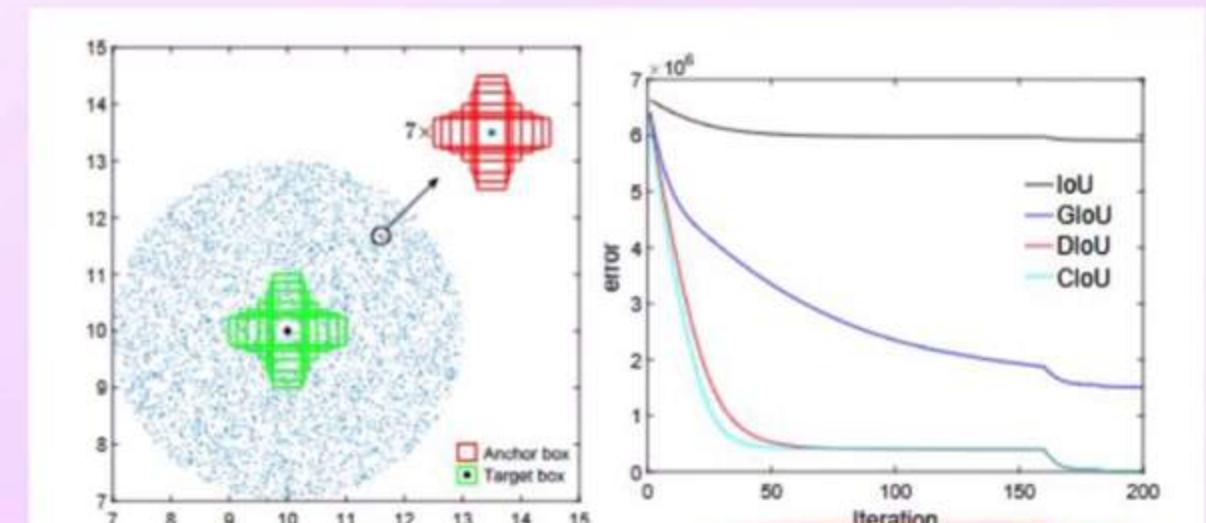


Figure 3: Simulation experiments: (a) 1,15,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

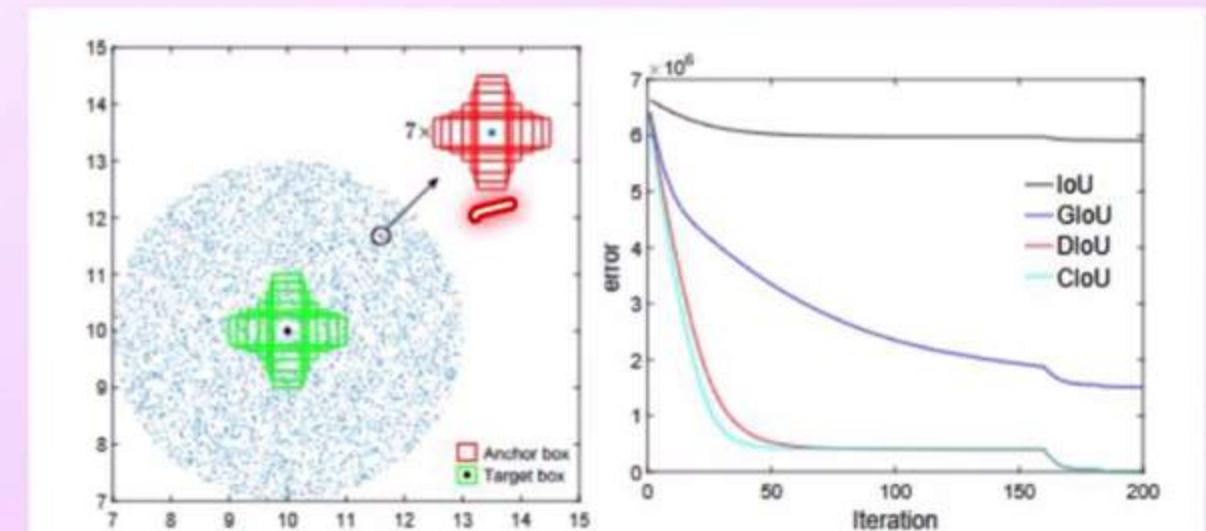


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

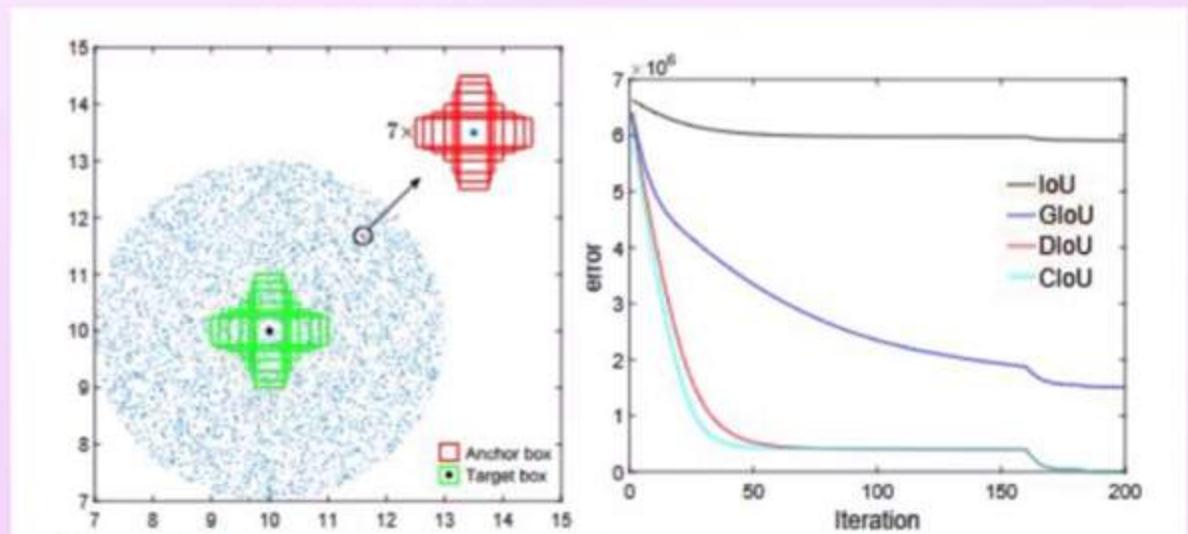


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

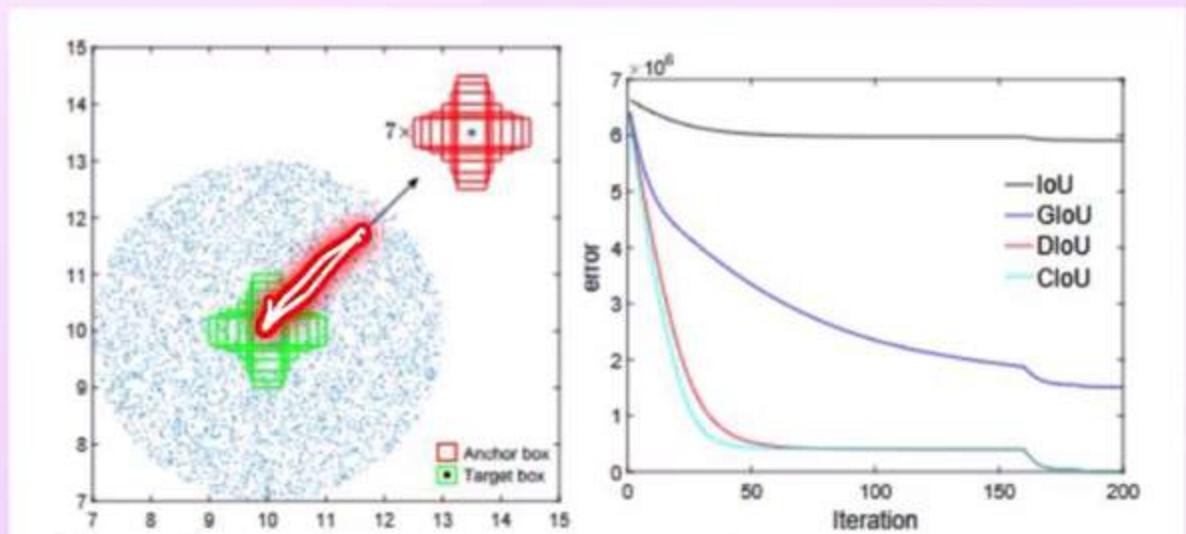


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

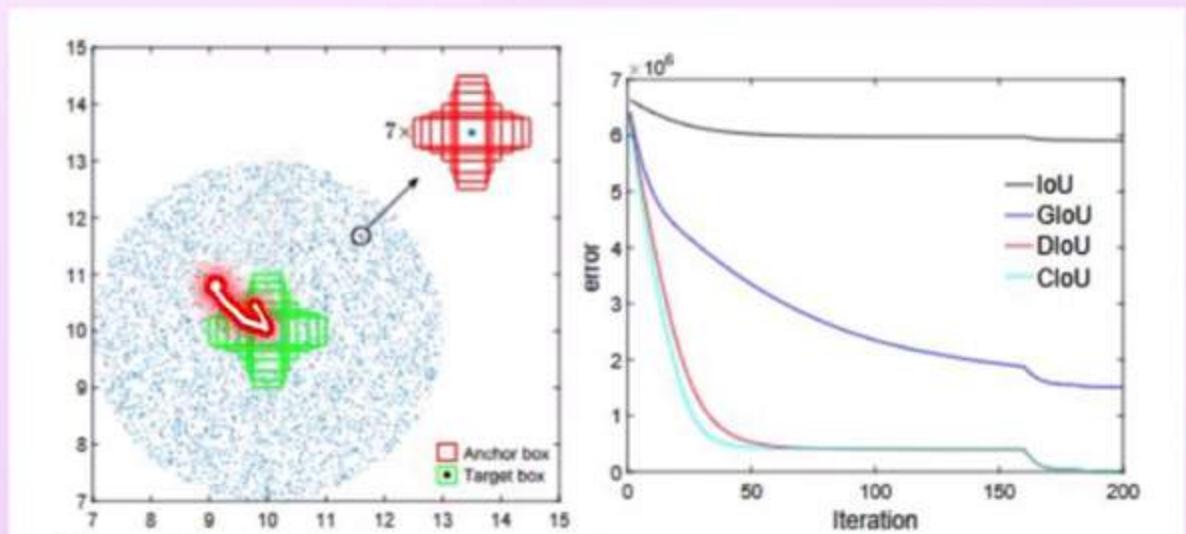


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

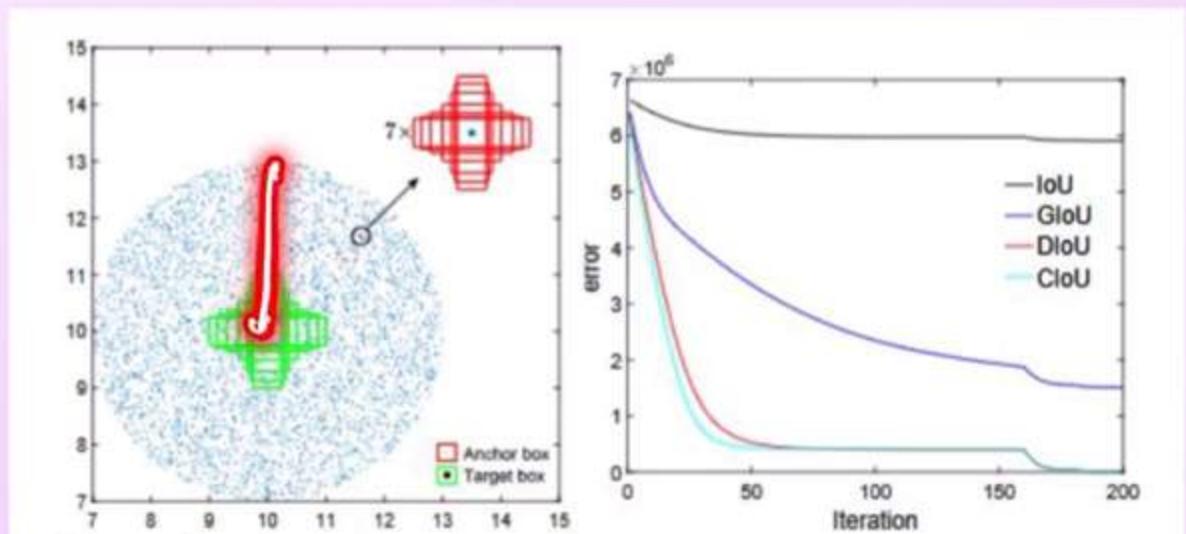


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

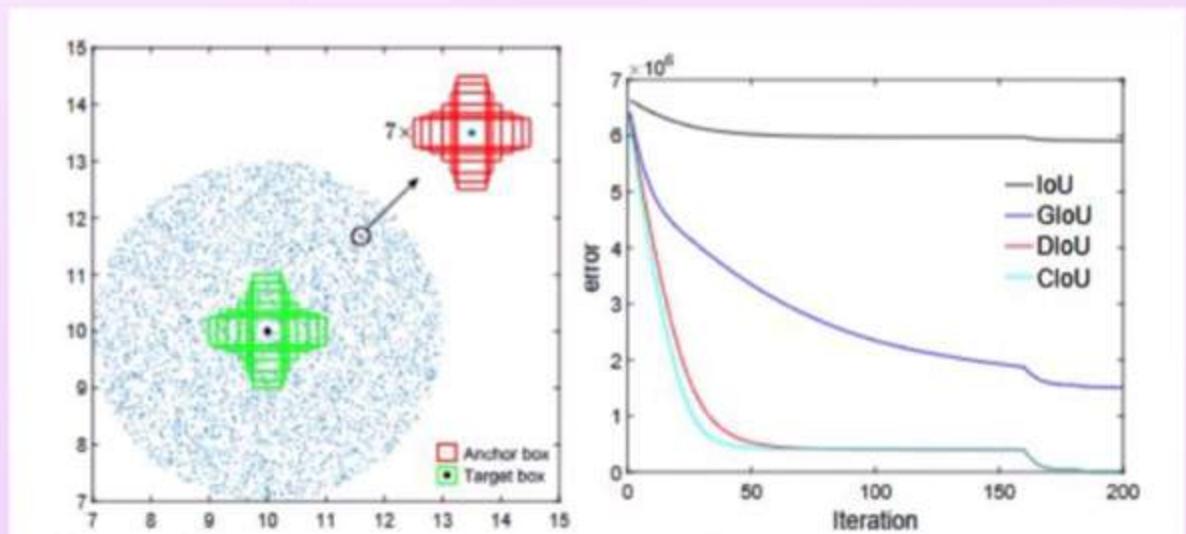


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

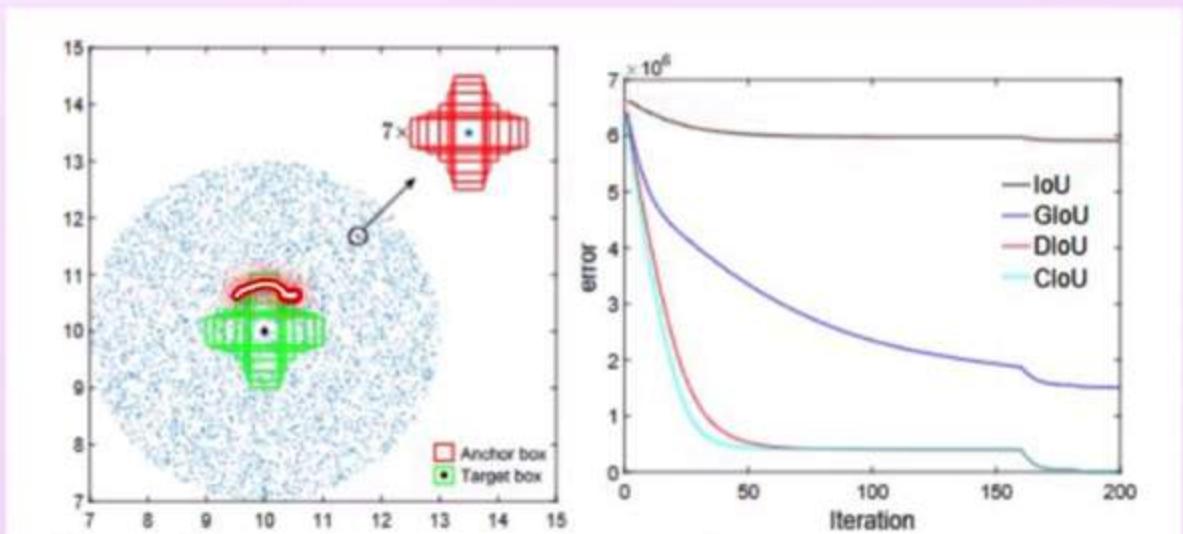


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

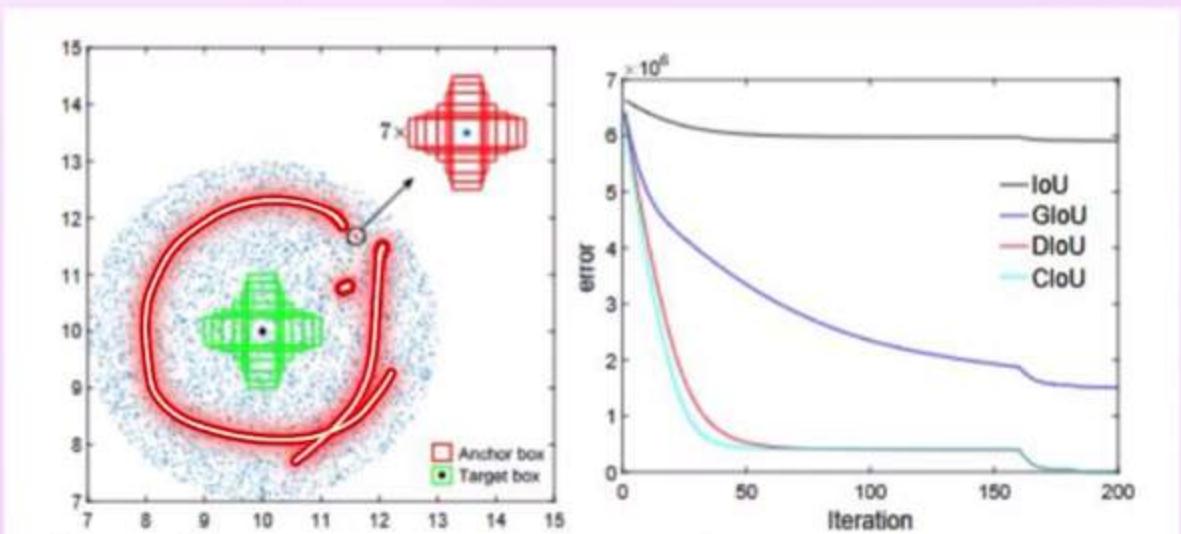


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

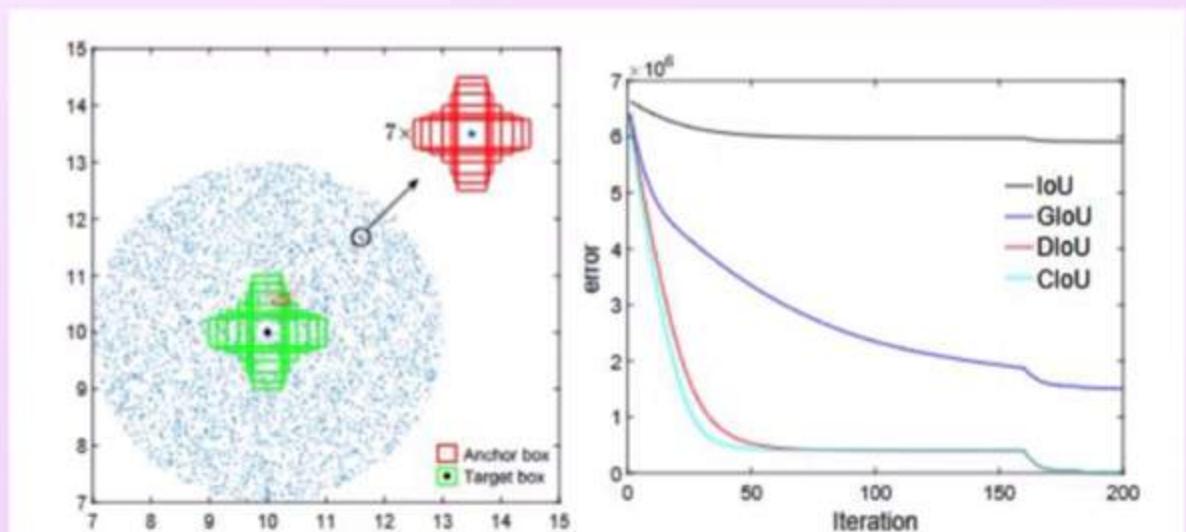


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

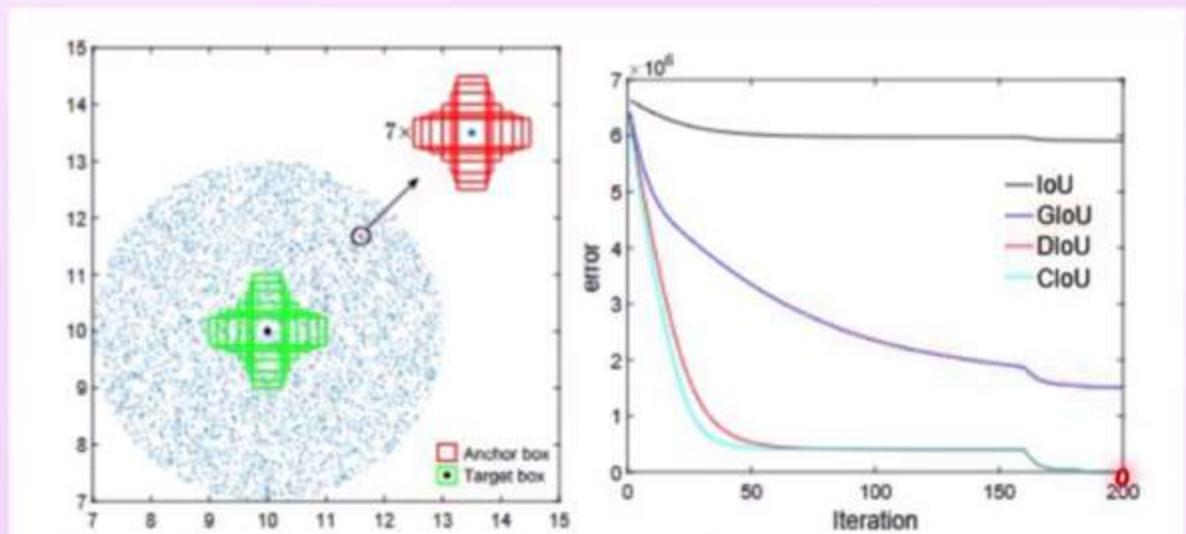


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

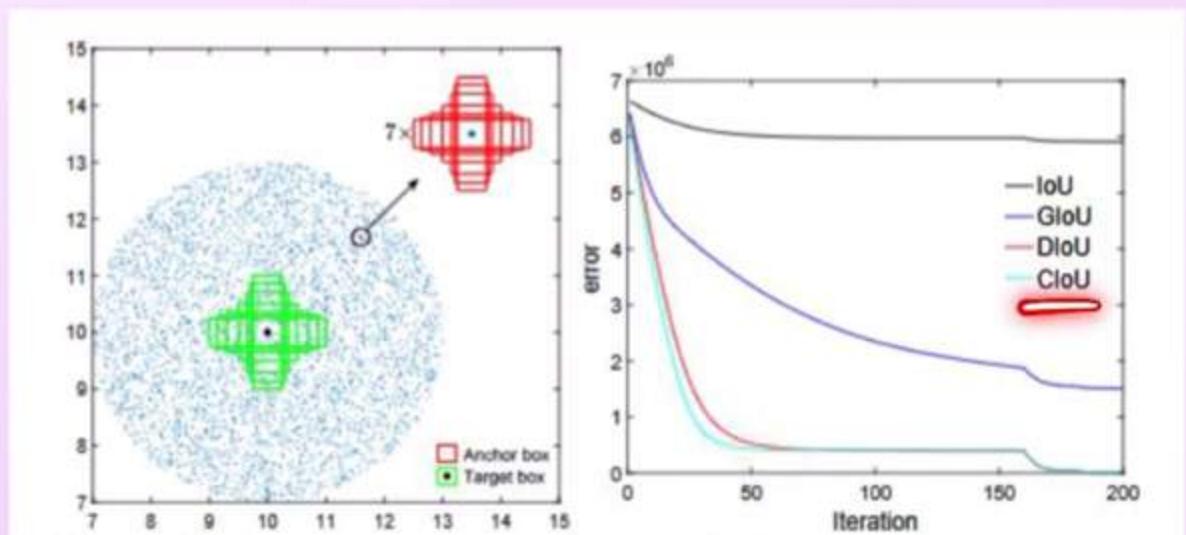


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# Complete IOU Loss

- Overlapping Area
- Distance between centers
- Aspect ratios

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v$$

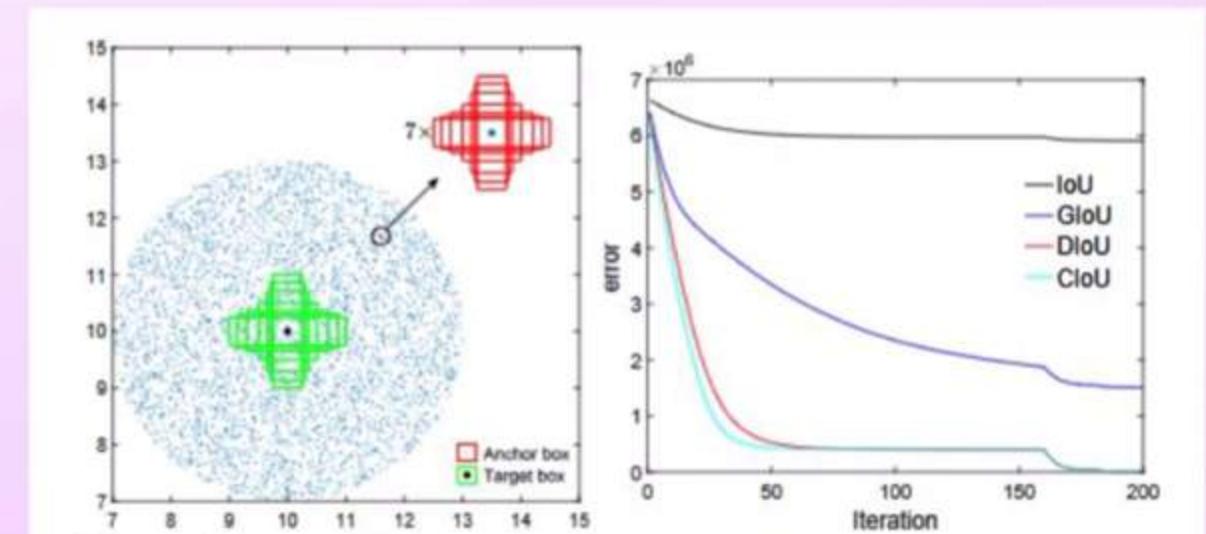


Figure 3: Simulation experiments: (a) 1,715,000 regression cases are adopted by considering different distances, scales and aspect ratios, (b) regression error sum (i.e.,  $\sum_n E(t, n)$ ) curves of different loss functions at iteration  $t$ .

# In this Video

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>   |

# In this Video

|                              | Backbone  | Detector   |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters →</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>   |

# In this Video

|                              | <b>Backbone</b>   | <b>Detector</b>  |
|------------------------------|---|--|
| <b>Bag of Freebies (BoF)</b> | <ul style="list-style-type: none"><li>• CutMix</li><li>• Mosaic data augmentation</li><li>• DropBlock</li><li>• Class label smoothing</li></ul> | <ul style="list-style-type: none"><li>• CloU-loss</li><li>• Cross mini-Batch Normalization</li><li>• DropBlock</li><li>• Mosaic data augmentation</li><li>• Self-Adversarial Training</li><li>• Multiple anchors for a single ground truth</li><li>• Cosine annealing scheduler</li><li>• Optimal hyperparameters</li><li>• Random training shapes</li></ul> |
| <b>Bag of Specials (BoS)</b> | <ul style="list-style-type: none"><li>• Multi-input weighted residual connections (MiWRC)</li></ul>   | <ul style="list-style-type: none"><li>• DIoU-NMS</li></ul>   |