ECCV TEL AVIV 2022

Learning to Weight Samples for Dynamic Early-exiting Networks

Yizeng Han[*1] , **Yifan Pu**[*1] , **Zihang Lai**[2†] , **Chaofei Wang**[1] , **Shiji Song**[1] , **Junfeng Cao**[3] , **Wenhui Huang**[3] , **Chao Deng**[3] , **and Gao Huang**[1]✉

* Equal contribution. ✉Corresponding author.

[1]Department of Automation, Tsinghua University.

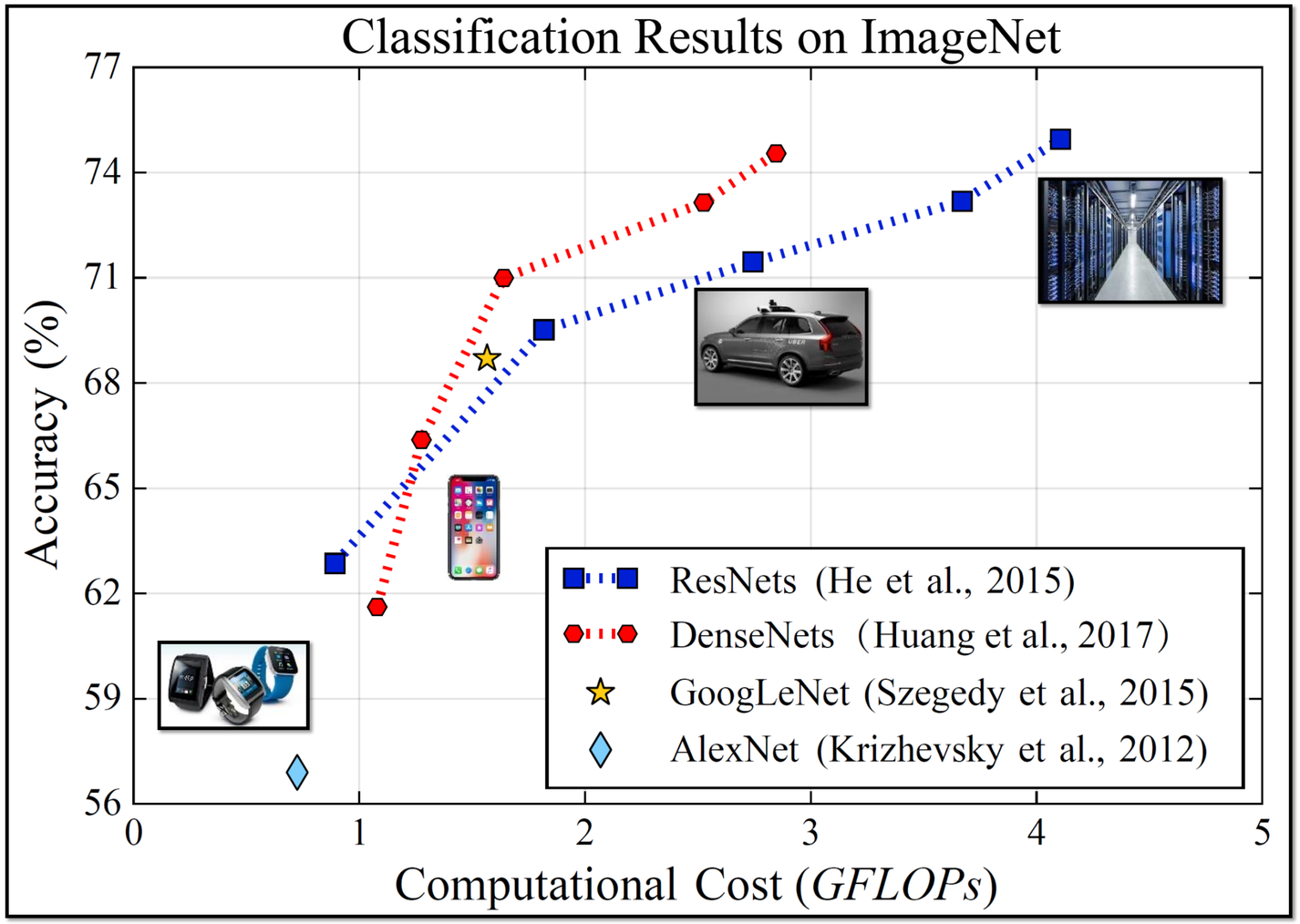
[2]Carnegie Mellon University. † Work done during an internship at Tsinghua University.

[3]China Mobile Research Institute.

中国移动 China Mobile

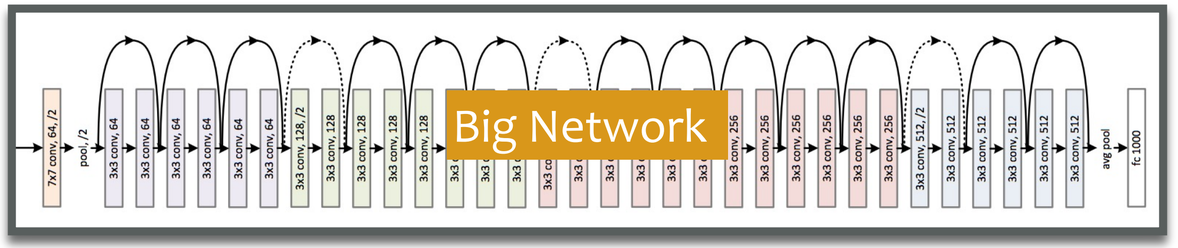
清華大學 Tsinghua University

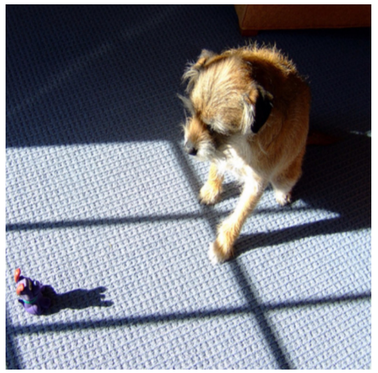
Accuracy-Efficiency Tradeoff



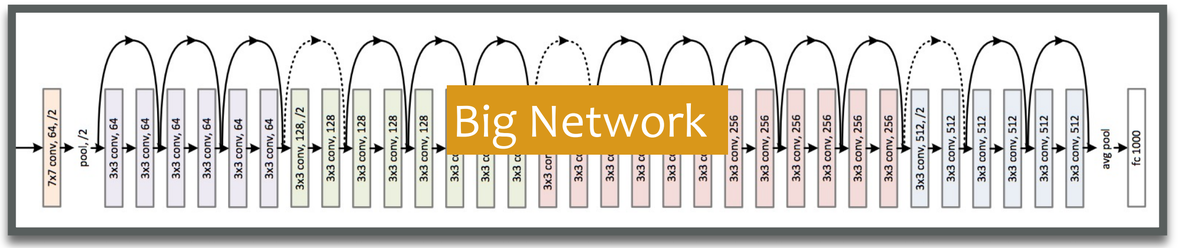
*Most conventional neural networks recognize all samples with **the same architecture**.*

Canonical (“easy”)



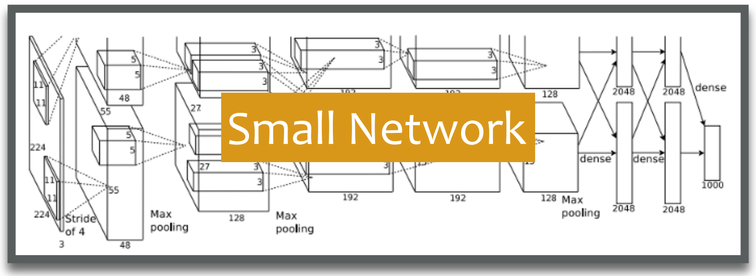


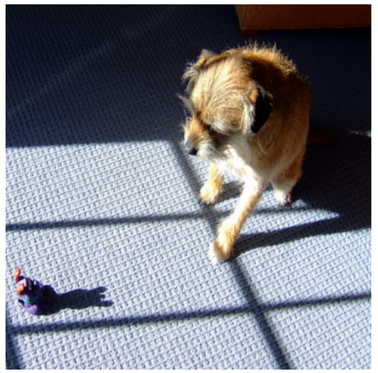
Noncanonical (“hard”)



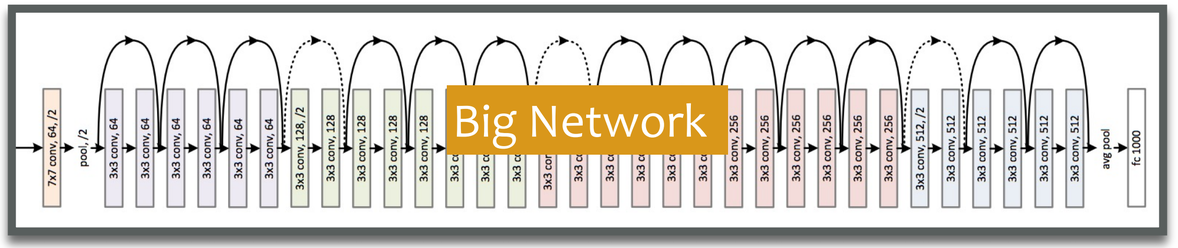
*A naïve idea of **dynamic** inference.*

Canonical (“easy”)

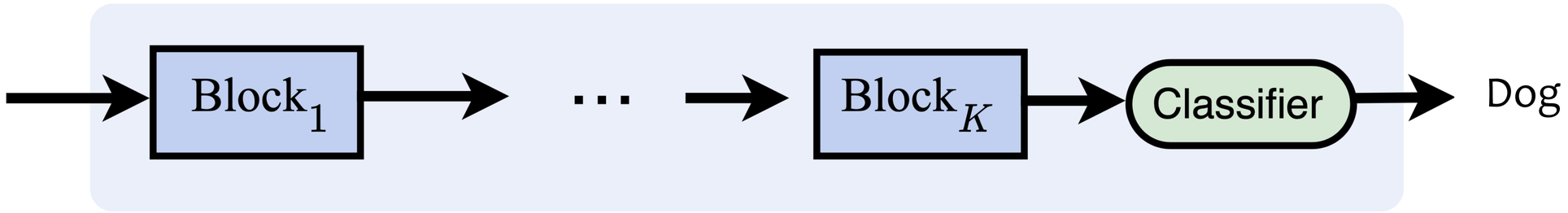




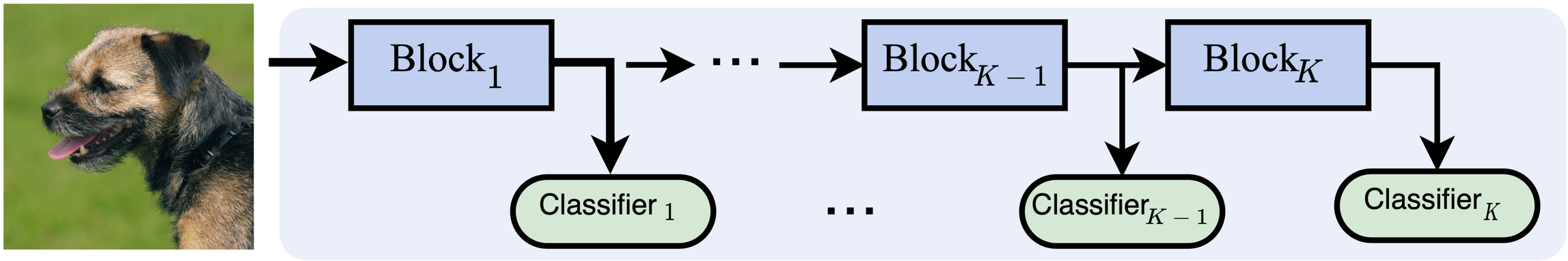
Noncanonical (“hard”)



Conventional inference procedure in a deep network



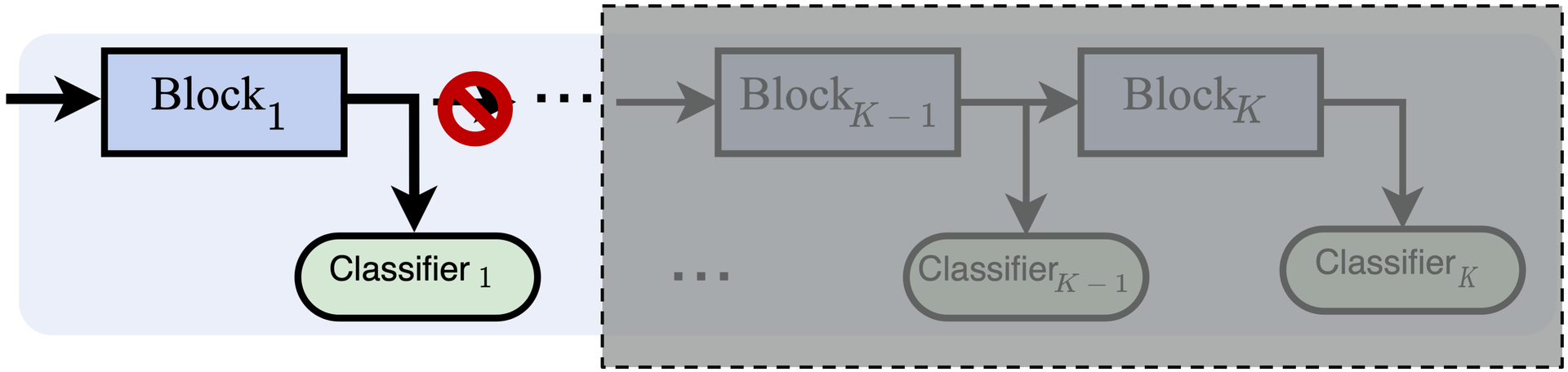
Dynamic early exiting[1] *in a multi-exit network.*



Prediction: Dog
Confidence = 0.9 > ε_1

[1] Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2021). Dynamic neural networks: A survey. IEEE TPAMI.

Dynamic early exiting[1] *in a multi-exit network.*

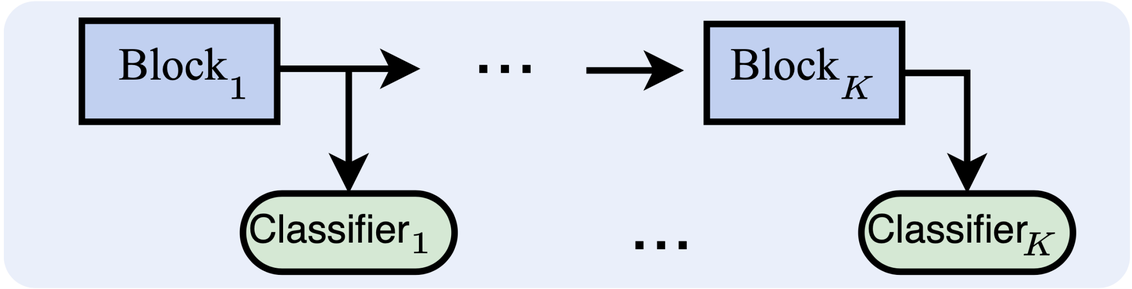


Not activated

Prediction: Dog
Confidence = 0.9 > ε_1

[1] Han, Y., Huang, G., Song, S., Yang, L., Wang, H., & Wang, Y. (2021). Dynamic neural networks: A survey. IEEE TPAMI.

Conventional training strategies[2,3,4] are sub-optimal to dynamic early exiting.



Inference Stage

"Easy" Samples

"Hard" Samples

Gap

Conventional Training


$1.0 \times \text{Loss}_{\text{easy}} + 1.0 \times \text{Loss}_{\text{hard}}$


$1.0 \times \text{Loss}_{\text{easy}} + 1.0 \times \text{Loss}_{\text{hard}}$

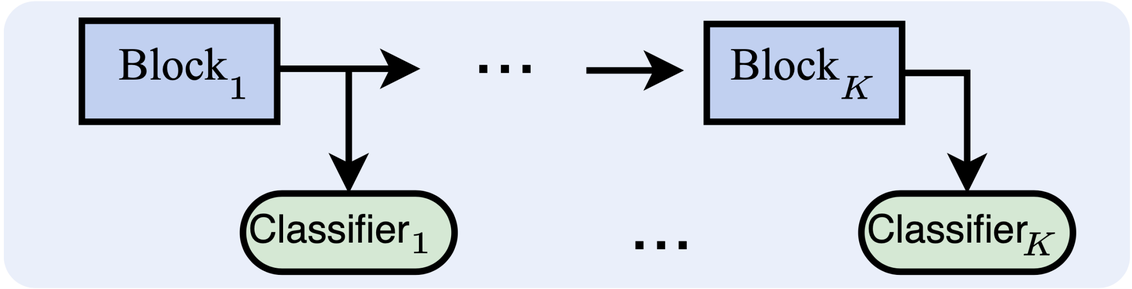
All training samples contribute equally to different exits (classifiers).

[2] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.
[3] Li, H., Zhang, H., Qi, X., Yang, R., & Huang, G. (2019). Improved techniques for training adaptive deep networks. In ICCV.
[4] Yang, L., Han, Y., Chen, X., Song, S., Dai, J., & Huang, G. (2020). Resolution adaptive networks for efficient inference. In CVPR.

***Our motivation**:*

*Training samples with **varying complexity** should **contribute differently** to the multiple exits in dynamic early-exiting networks.*

We bridge the gap between training and testing by ***sample weighting***



Inference Stage

"Easy" Samples

"Hard" Samples

Gap

Conventional Training


$1.0 \times \text{Loss}_{\text{easy}} + 1.0 \times \text{Loss}_{\text{hard}}$


$1.0 \times \text{Loss}_{\text{easy}} + 1.0 \times \text{Loss}_{\text{hard}}$

Our Training


$1.2 \times \text{Loss}_{\text{easy}} + 0.8 \times \text{Loss}_{\text{hard}}$


$0.6 \times \text{Loss}_{\text{easy}} + 1.4 \times \text{Loss}_{\text{hard}}$

Sample weighting formulation

- f: multi-exit dynamic network.
- $f^{(k)}\left(\cdot;\boldsymbol{\Theta}_f^{(k)}\right)$: the k-th sub-network parameterized by $\boldsymbol{\Theta}_f^{(k)}$.

- Traditional training objective:

$$\mathcal{L}=\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N} \quad l_i^{(k)}$$

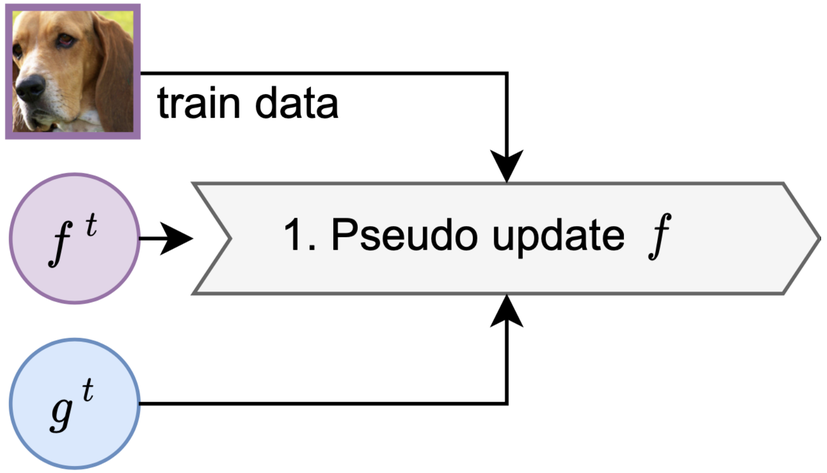
- Our modified objective with sample weighting:

$$\mathcal{L}=\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N} w_i^{(k)} l_i^{(k)}$$

- $\boldsymbol{w}_i$ is obtained from weight prediction network g: $\boldsymbol{w}_i=g\left(\boldsymbol{l}_i;\boldsymbol{\Theta}_g\right)$,
  where $\boldsymbol{w}_i=\left[w_i^{(1)},w_i^{(2)},\dots,w_i^{(K)}\right]$,
  $\boldsymbol{l}_i=\left[l_i^{(1)},l_i^{(2)},\dots,l_i^{(K)}\right]$

Meta-learning: *training pipeline* [5]

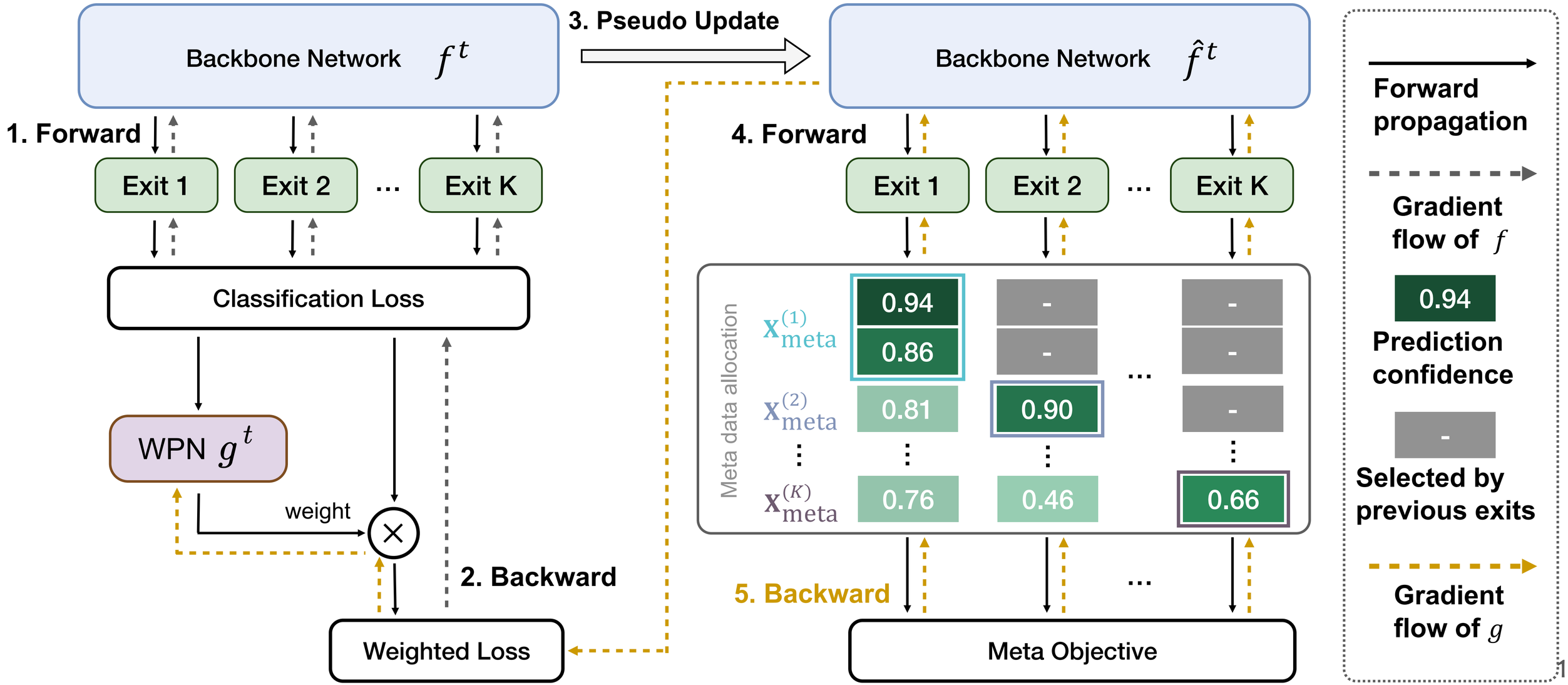
- $f(\cdot;\mathbf{\Theta}_f)$: multi-exit classification network.
- $g\left(\cdot;\mathbf{\Theta}_g\right)$: weight prediction network



We re-use the training dataset as our meta-dataset rather than using a standalone meta-dataset as in [5].

[5] Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., & Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. In NeurIPS.

Meta-learning: *optimization*



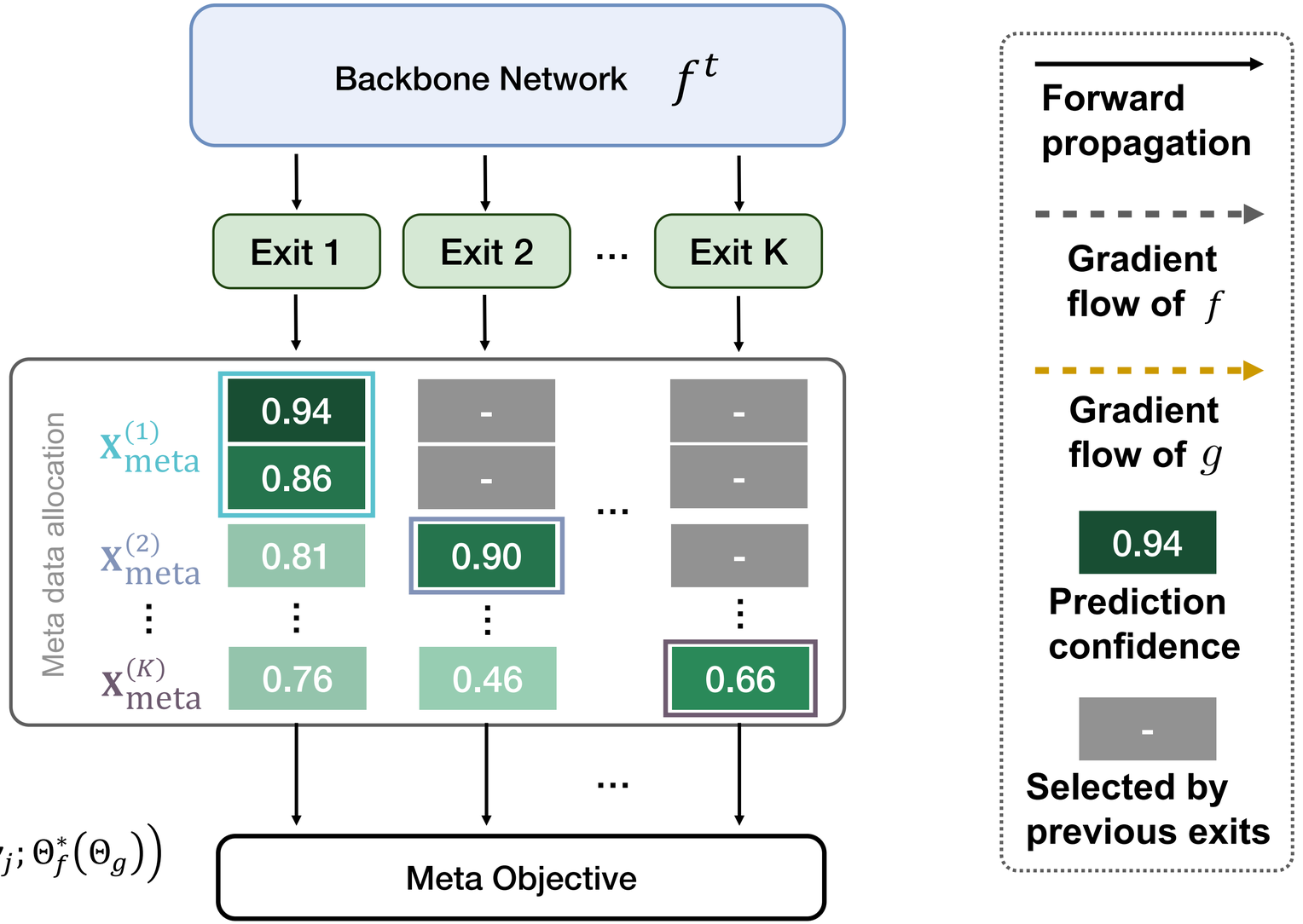
Meta-data allocation

$$\left|\mathbf{X}_{\text{meta}}^{(k)}\right| = \frac{q^k}{\sum_{k=1}^{K} q^k} N_{\text{meta}}, \qquad q > 0.$$

$q = 1.0$: uniform distribution

$$\Theta_g^* = \underset{\Theta_g}{\operatorname{argmin}} \mathcal{L}_{\text{meta}}\left(\Theta_f^*(\Theta_g)\right)$$

$$\triangleq \underset{\Theta_g}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{j:\mathbf{x}_j \in \mathbf{X}_{\text{meta}}^{(k)}} l^{(k)}\left(\mathbf{x}_j, y_j; \Theta_f^*(\Theta_g)\right)$$



Meta-learning objectives

Weighted CE Loss: optimizing the classification network f

$$\Theta_f^*(\Theta_g) = \underset{\Theta_f}{\operatorname{argmin}}\, \mathcal{L}_{\text{tr}}(\Theta_f, \Theta_g)$$

$$\triangleq \underset{\Theta_f}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{N} \sum_{i:\mathbf{x}_i \in \mathbf{X}_{\text{tr}}} g^{(k)}(\boldsymbol{l}_i; \Theta_g) \cdot l_i^{(k)}(\Theta_f).$$

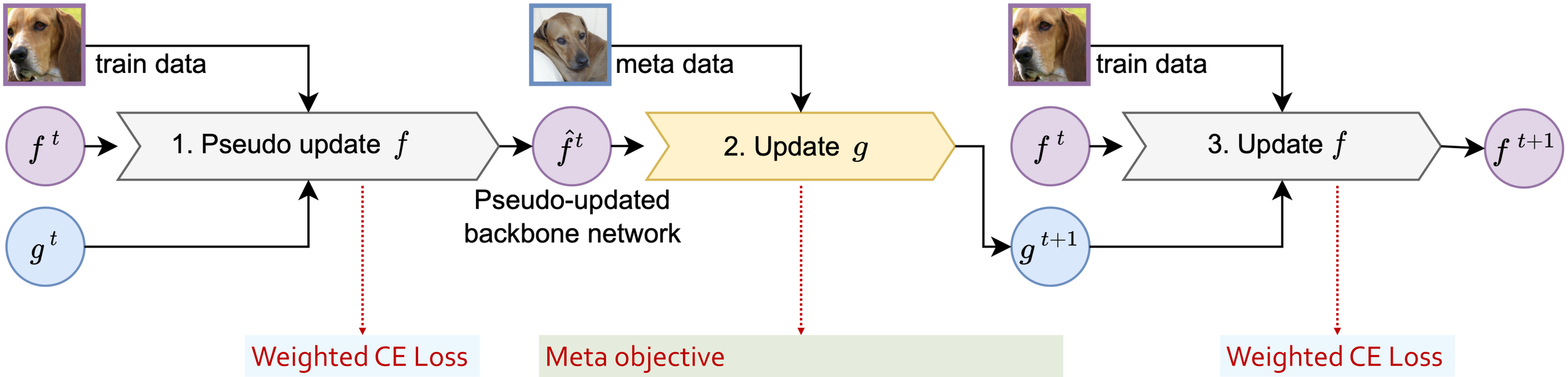
Meta objective: optimizing the weight prediction network g

$$\Theta_g^* = \underset{\Theta_g}{\operatorname{argmin}}\, \mathcal{L}_{\text{meta}}\left(\Theta_f^*(\Theta_g)\right)$$

$$\triangleq \underset{\Theta_g}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{j:\mathbf{x}_j \in \mathbf{X}_{\text{meta}}^{(k)}} l^{(k)}\left(\mathbf{x}_j, y_j; \Theta_f^*(\Theta_g)\right).$$

Method: *summary*

- $f(\cdot;\boldsymbol{\Theta}_f)$: multi-exit classification network.
- $g\left(\cdot;\boldsymbol{\Theta}_g\right)$: weight prediction network



Experiments

- Model
  - Multi-scale Dense Network (MSDNet) [2]
  - Resolution Adaptive Network (RANet) [4]

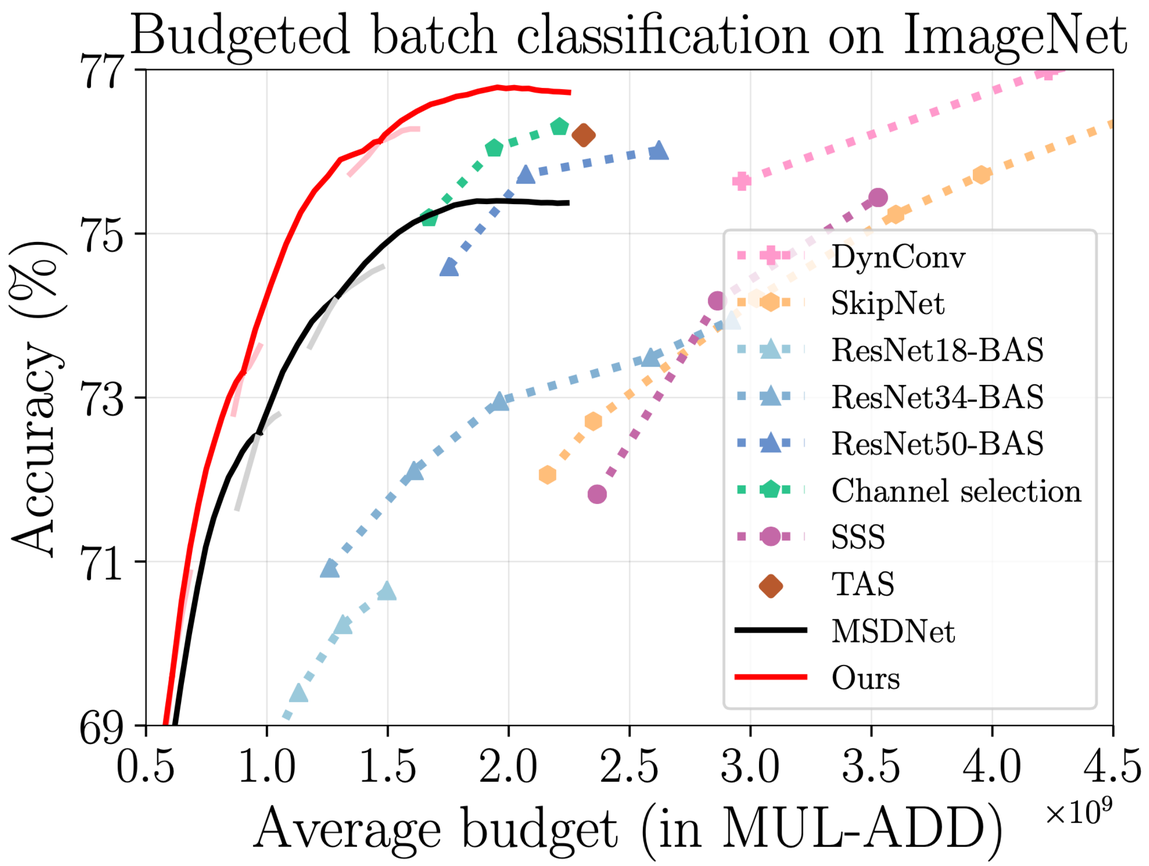
- Dataset
  - ImageNet
  - CIFAR-10, CIFAR-100
  - Long-tailed CIFAR-100 [6]

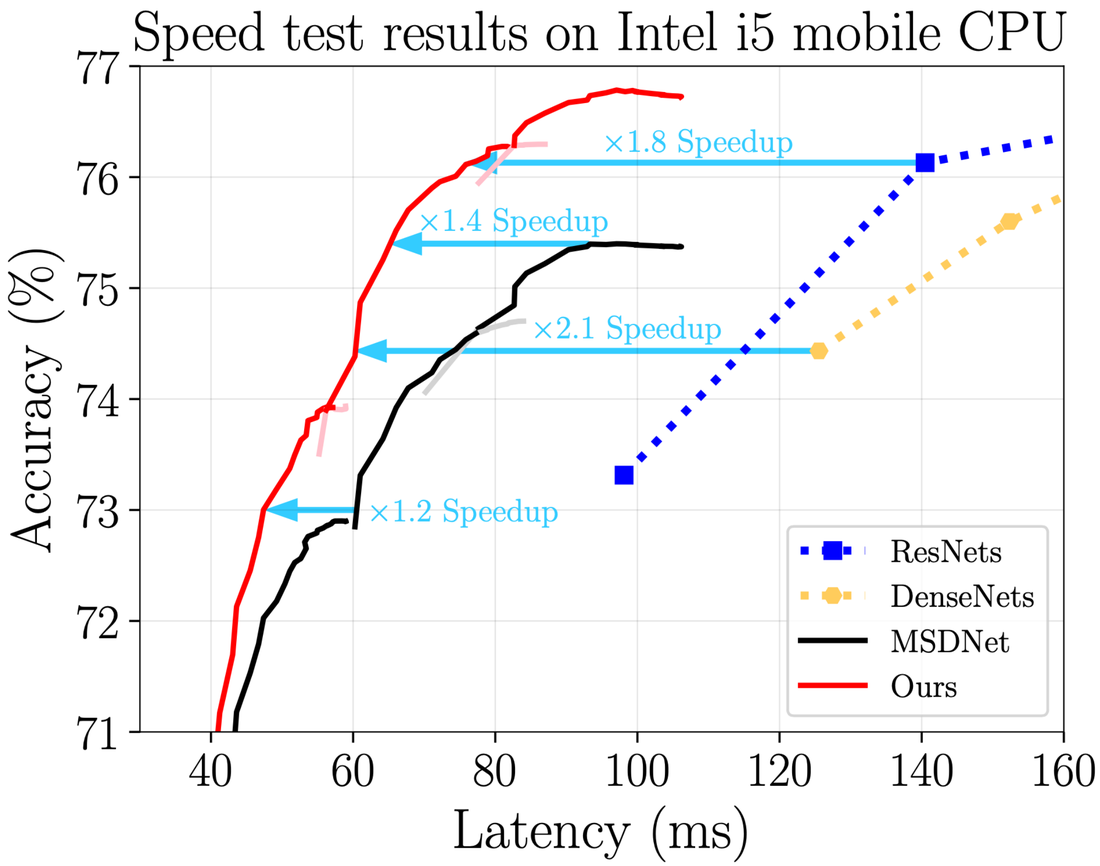
- Other training techniques [3]

[2] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.
[3] Li, H., Zhang, H., Qi, X., Yang, R., & Huang, G. (2019). Improved techniques for training adaptive deep networks. In ICCV.
[4] Yang, L., Han, Y., Chen, X., Song, S., Dai, J., & Huang, G. (2020). Resolution adaptive networks for efficient inference. In CVPR.
[6] Cui, Y., Jia, M., Lin, T. Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In CVPR

Experiments:

Results on ImageNet with MSDNet [2]

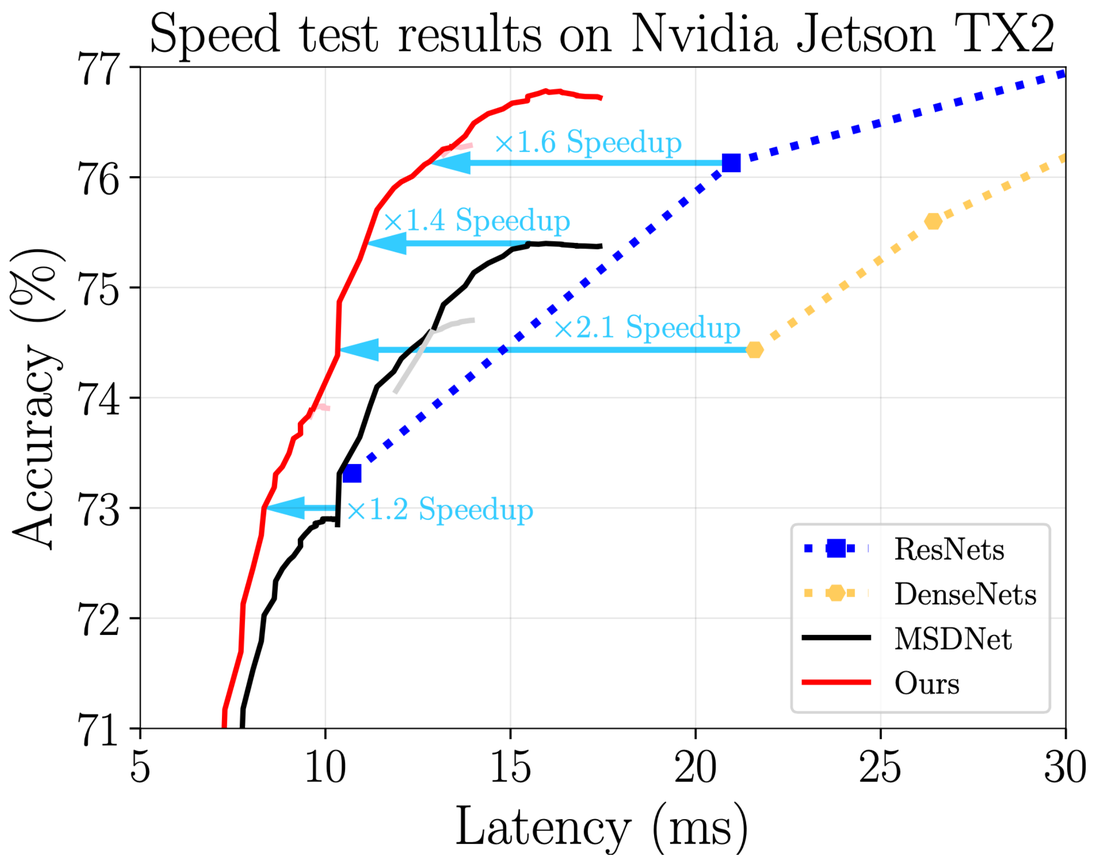


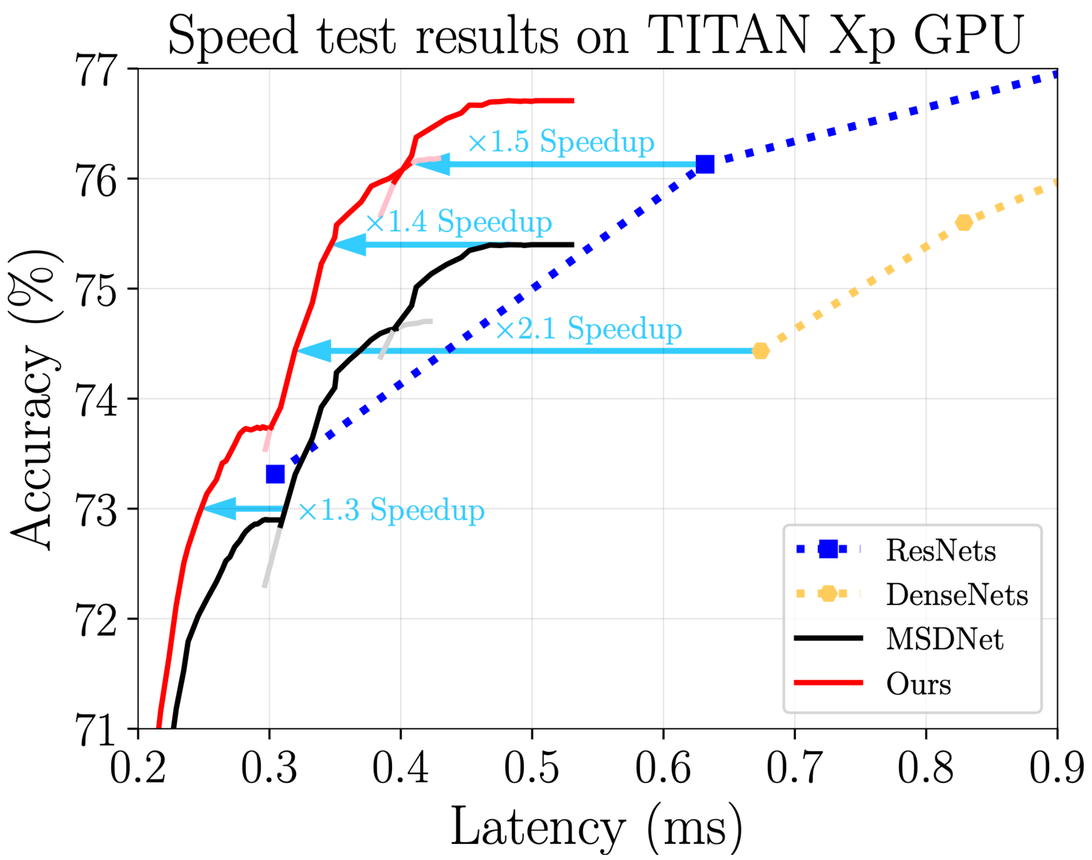


[2] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.

Experiments:

Results on ImageNet with MSDNet [2]

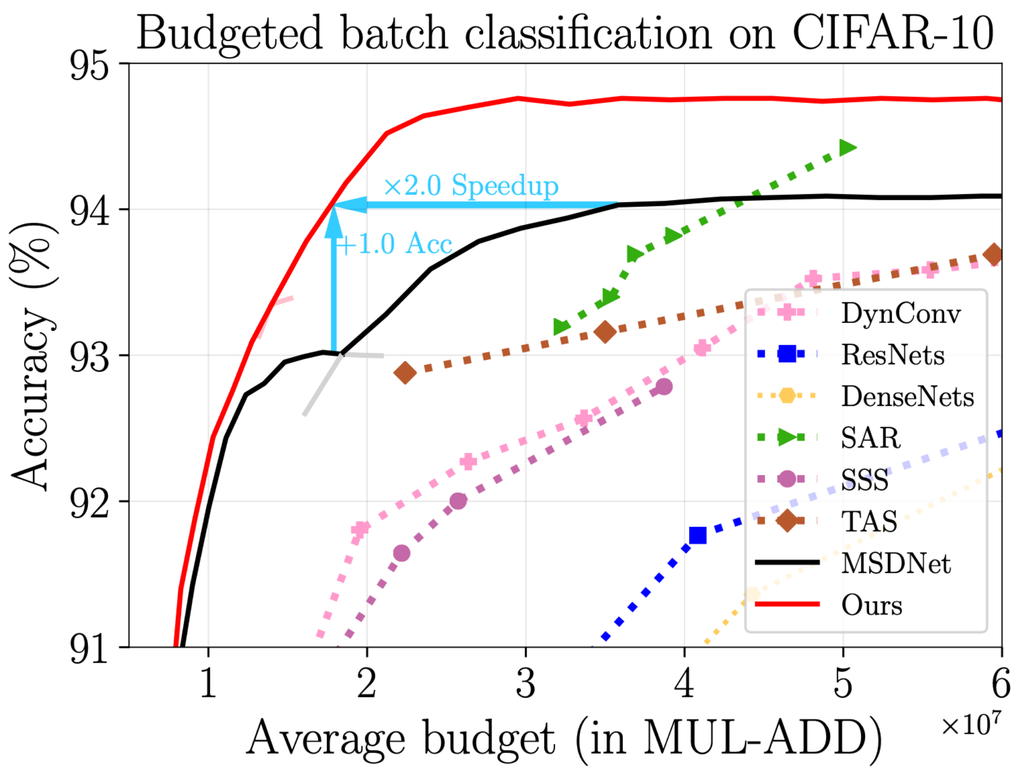


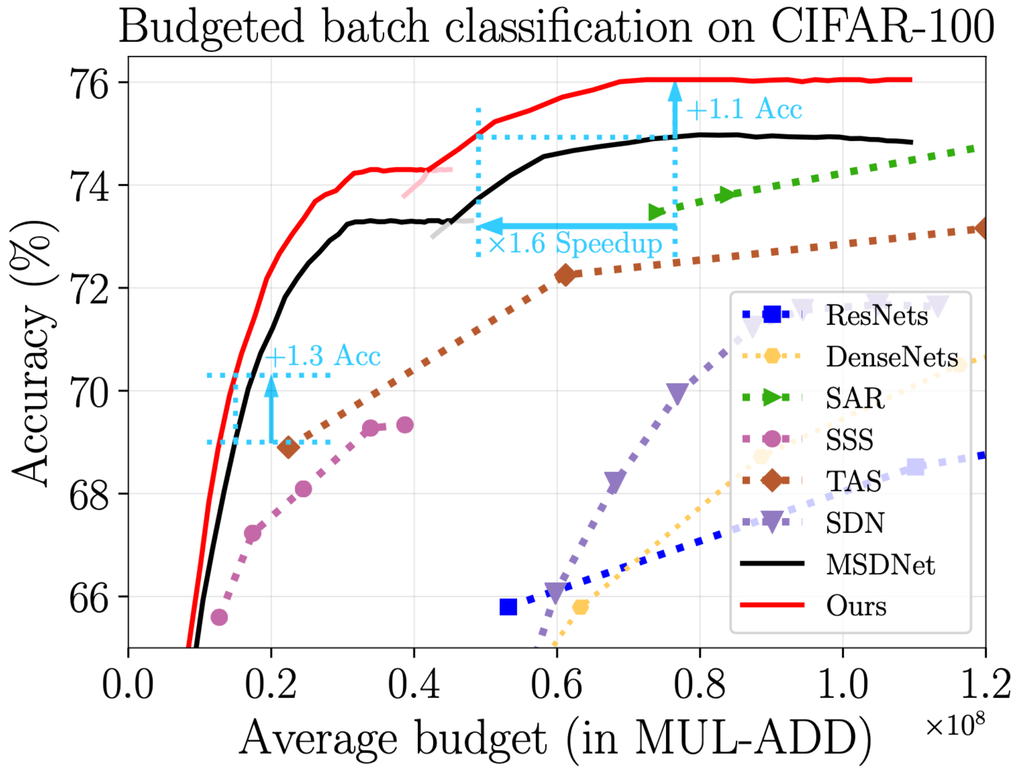


[2] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.

Experiments:

CIFAR results with MSDNet [2]

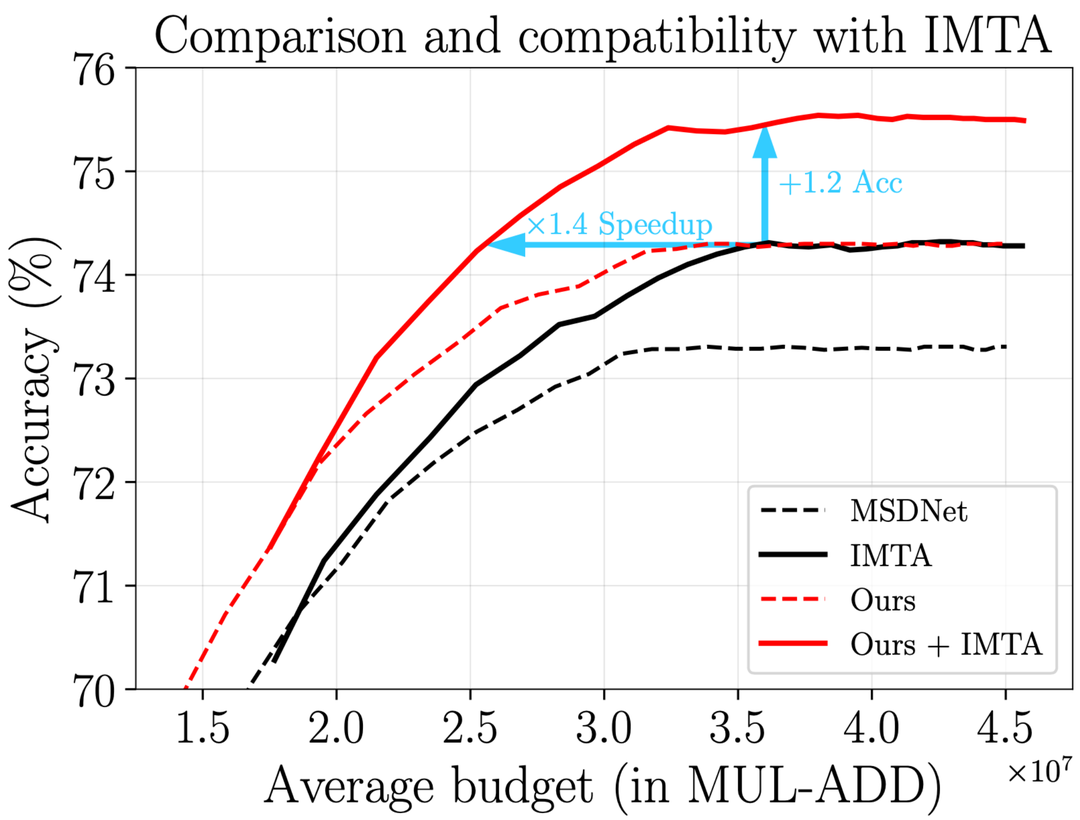


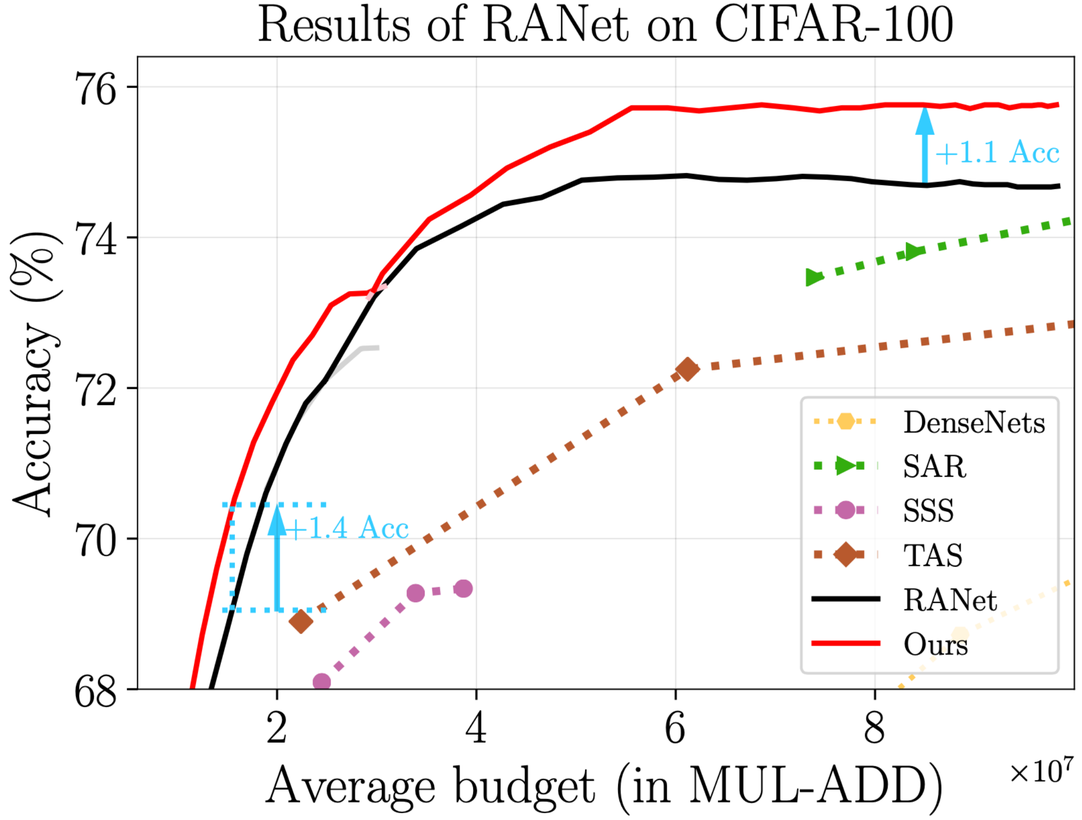


- Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.

Experiments:

Comparison with IMTA [3] and Results with RANet [4]



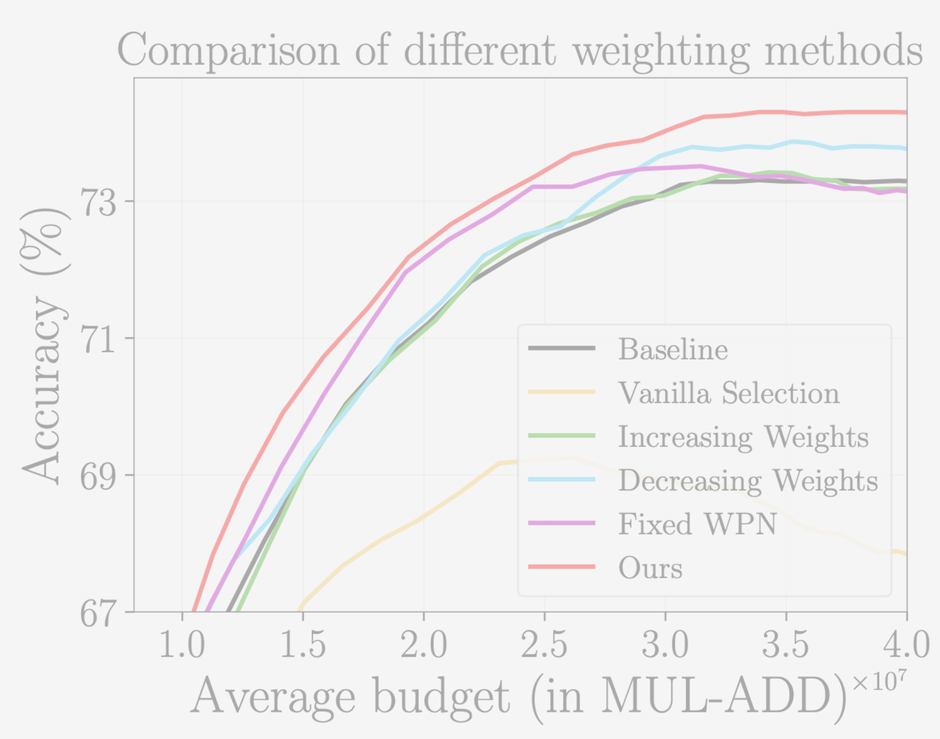


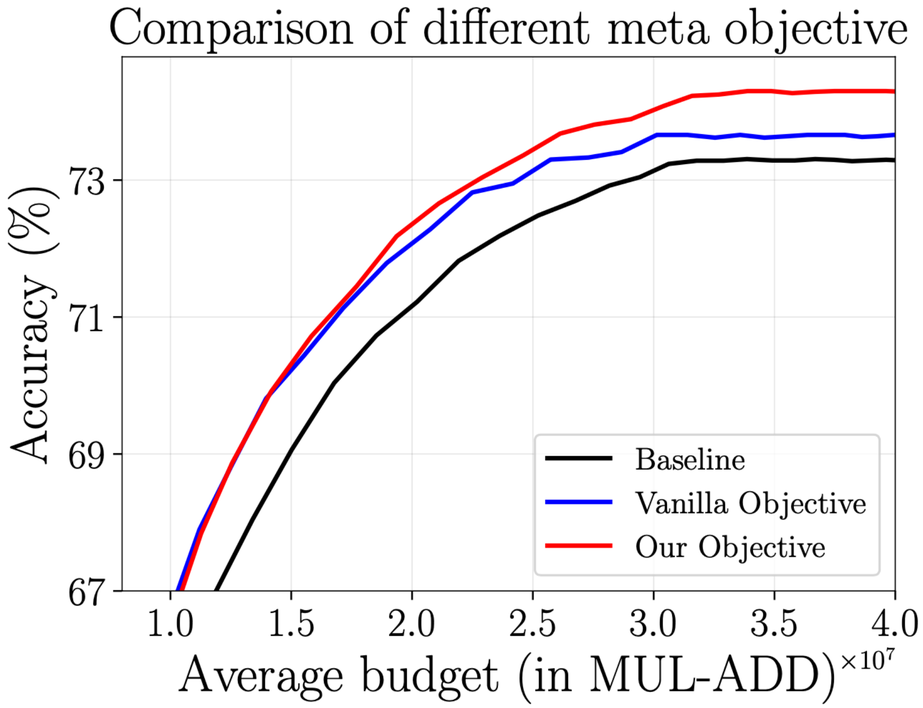
[3] Li, H., Zhang, H., Qi, X., Yang, R., & Huang, G. (2019). Improved techniques for training adaptive deep networks. In ICCV.
[4] Yang, L., Han, Y., Chen, X., Song, S., Dai, J., & Huang, G. (2020). Resolution adaptive networks for efficient inference. In CVPR.

Experiments:

ablation studies on CIFAR-100 with MSDNet [1]

- **Sample weighting** *v.s.* **Training data allocation**
- **Predicted weights** *v.s.* **fixed weights**
- **Meta-learning** *v.s.* **frozen weight prediction network**
- **Our meta objective** *v.s.* **vanilla meta objective**

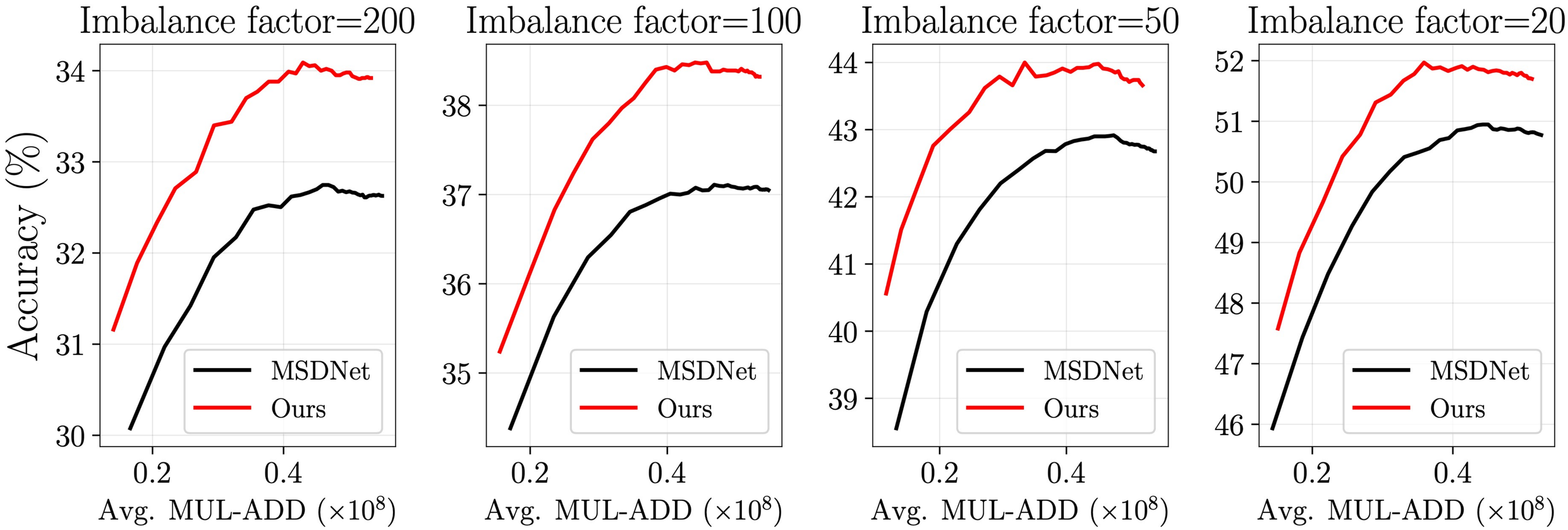




[1] Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. (2018, February). Multi-Scale Dense Networks for Resource Efficient Image Classification. In ICLR.

Experiments:

Results in the class-imbalanced setting



Conclusion (takeaway messages)

- **[Observation]** Different exits (classifiers) in dynamic early exiting networks are responsible for samples with **different complexity** at test time.
- **[Intuition]** Different samples (with varying **complexity**) should contribute **unequally** to different exits (with varying **capacity**) in training.
- **[Solution]** The gap between training and test can be bridged by **sample weighting** rather than sample selection (data allocation).
- **[Key design]** **Sample selection (data allocation)** in calculating the **meta objective** can successfully guide the learning of weight prediction.

ECCV TEL AVIV 2022

Thank you!
Welcome to our poster

Paper

Code

中国移动 China Mobile

清華大學 Tsinghua University