Loss Function Discovery for Object Detection via Convergence-Simulation Driven Search

Peidong Liu[1*], Gengwei Zhang[2*], Bochao Wang[3], Hang Xu[3], Xiaodan Liang[2], Yong Jiang[1], Zhenguo Li[3]

[1]Tsinghua University, [2]Sun Yat-Sen University, [3]Huawei Noah's Ark Lab

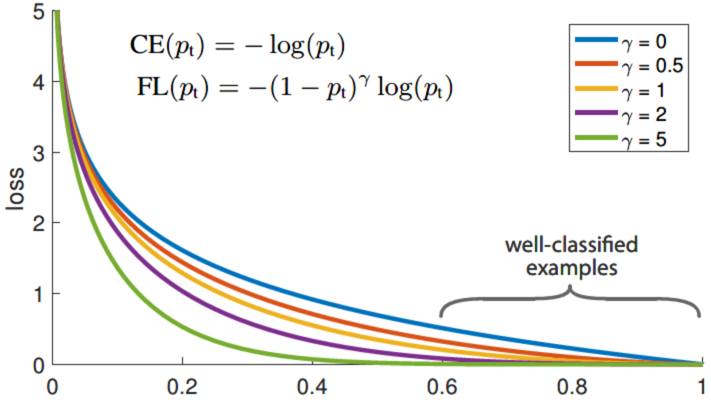
Reporter: **Peidong Liu**
Homepage: http://perdonliu.github.io/

CONTENTS

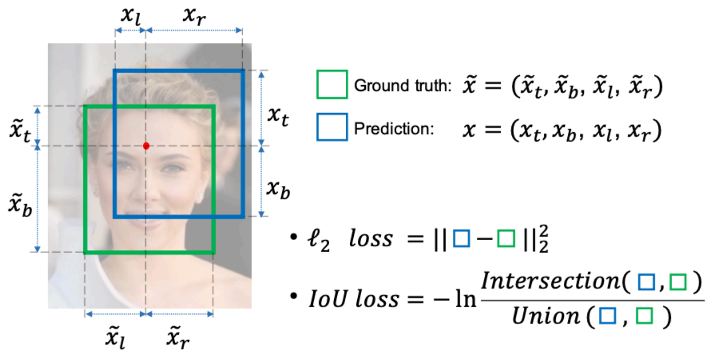
INTRODUCTION

Motivation

- Handcrafted loss functions are sub-optimal, e.g. CE loss, Focal loss[1], L1, IoU loss[2], GIoU loss[3]



Focal Loss[1]



IoU loss[2]

- Directly search loss formulas from well-designed primitive operations, different from searching hyperparameters in fixed loss formula[4, 5]

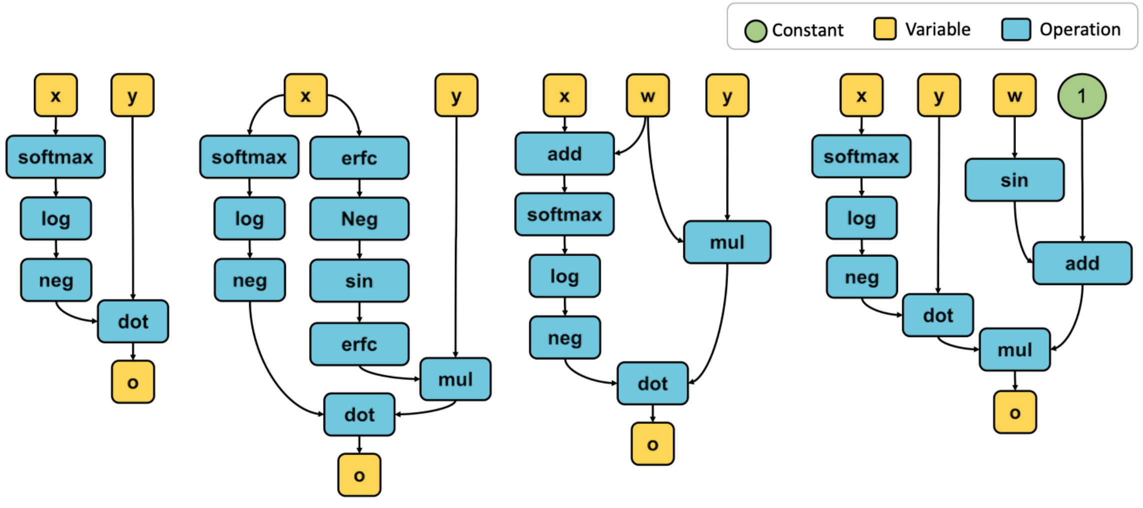
- Better alignment with evaluation metric (e.g. GFocal loss[6])

- Good mathematical property and optimization behavior

Key Challenges

- Alignment with evaluation metric (e.g. AP) → Well-designed Search Space
- Sparse action space (1 acceptable in 10^5)
- heavy evaluation time (around 30min for one loss function)
- Guarantee good mathematical property and optimization behavior

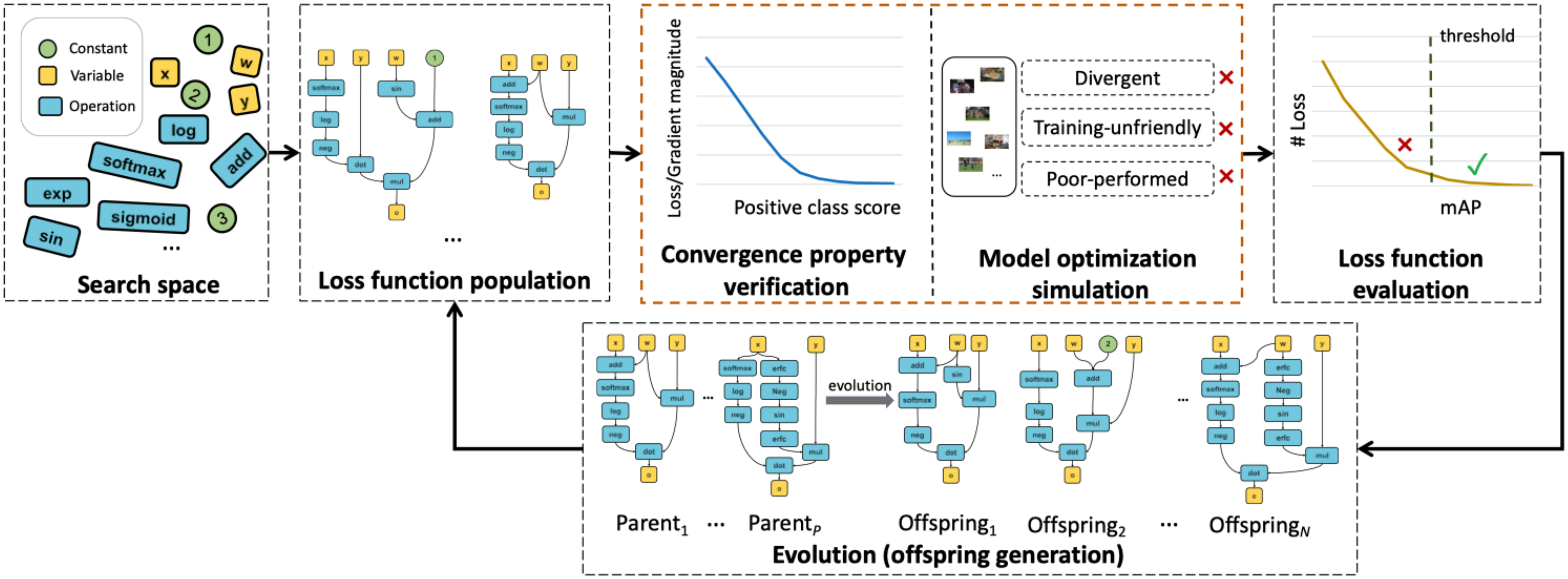
Convergence-Simulation Modules



AP: 37.6% (a) Cross-entropy loss | AP: 37.7% (b) Candidate1 | AP: 37.8% (c) Candidate2 | AP: 38.2% (d) CSE-Autoloss-A_{cls}

METHOD

Pipeline


Constant
Variable
Operation
log
softmax
add
exp
sigmoid
sin
Search space
Loss function population
Loss/Gradient magnitude
Positive class score
Convergence property verification
Divergent
Training-unfriendly
Poor-performed
Model optimization simulation
threshold
Loss
mAP
Loss function evaluation
evolution
Parent1 … ParentP
Offspring1 Offspring2 … OffspringN
Evolution (offspring generation)

Search Space Design

- Input Nodes

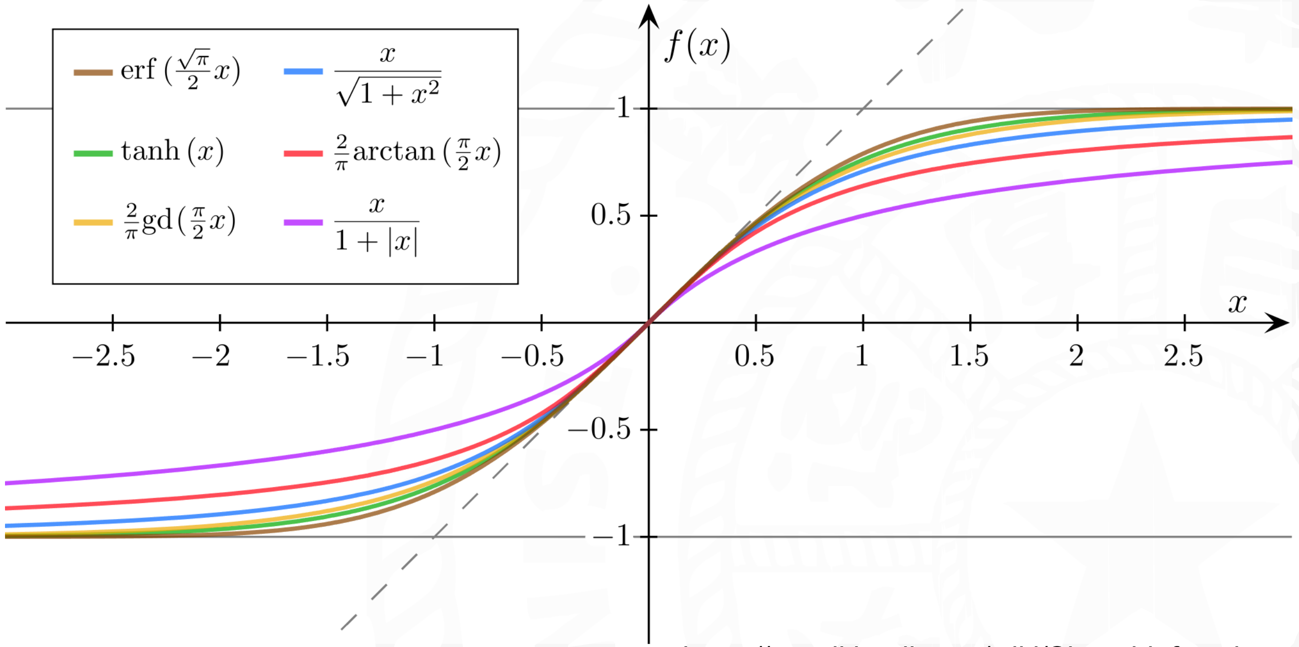
Classification: prediction x, label y, IoU w
Regression: intersection i, union u, closure e

$$CE(x,y) = -dot(y, log(softmax(x)))$$
$$CEI(x,y,w) = -dot(wy, log(softmax(x)))$$

Detector	Loss	AP (%)
Faster R-CNN R50	CE + GIoU	37.6
	CEI + GIoU	**37.7**$^{+0.1}$

- Primitive Operators

- Unary functions: $-x$, e^x, $\log(x)$, $|x|$, $\sqrt{x}$, softmax(x), softplus(x), $\sigma(x)$, gd(x), alf(x), erf(x), erfc(x), tanh(x), relu(x), sin(x), cos(x)
- Binary functions: $x_1 + x_2$, $x_1 - x_2$, $x_1 \times x_2$, $\frac{x_1}{x_2+\epsilon}$, dot(x_1, x_2)



https://en.wikipedia.org/wiki/Sigmoid_function

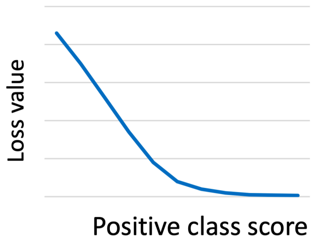
Convergence-Simulation Modules

■ Convergence Property Verification

guarantee *monotonicity* and *convergence*

Classification Branch

$$\mathrm{BCE}(x) = -\ln\left(\frac{1}{1+e^{-x}}\right), \quad \frac{\partial \mathrm{BCE}(x)}{\partial x} = -1 + \frac{1}{1+e^{-x}},$$

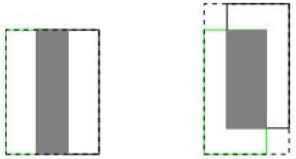


Convergence property verification

- Monotonicity
- Convergence

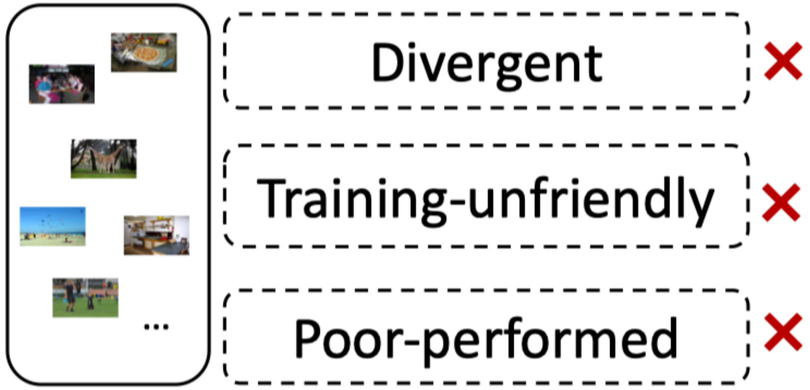
Regression Branch

- Distance-loss consistency



■ Model Optimization Simulation

guarantee *training-friendliness* and *good performance*



Simulate the optimization behavior of the loss candidates with *#class* images

Model optimization simulation

EXPERIMENTS

Search on COCO

- Faster R-CNN R50

$$\text{CSE-Autoloss-A}_{\text{cls}}(x, y, w) = -\text{dot}((1 + \sin(w))y, \log(\text{softmax}(x))),$$
$$\text{CSE-Autoloss-A}_{\text{reg}}(i, u, e) = (1 - \frac{i}{u}) + (1 - \frac{i+2}{e}),$$

- FCOS R50

$$\text{CSE-Autoloss-B}_{\text{cls}}(x, y, w) = -[wy(1 + \text{erf}(\sigma(1 - y)))\log\sigma(x) + (\text{gd}(x) - wy)(\sigma(x) - wy)\log(1 - \sigma(x))],$$
$$\text{CSE-Autoloss-B}_{\text{reg}}(i, u, e) = \frac{3eu + 12e + 3i + 3u + 18}{-3eu + iu + u^2 - 15e + 5i + 5u}.$$

Search on COCO

- Across Architectures

Detector	Loss	AP (%)	AP_{50}(%)	AP_{75}(%)	AP_S(%)	AP_M(%)	AP_L(%)
Faster R-CNN R50	CE + L1	37.4	58.1	40.4	21.2	41.0	48.1
	CE + GIoU	$37.6^{+0.2}$	$58.2^{+0.1}$	$41.0^{+0.6}$	$21.5^{+0.3}$	$41.1^{+0.1}$	$48.9^{+0.8}$
	CSE-Autoloss-A	$\mathbf{38.5^{+1.1}}$	$58.6^{+0.5}$	$41.8^{+1.4}$	$22.0^{+0.8}$	$42.2^{+1.2}$	$50.2^{+2.1}$
Faster R-CNN R101	CE + L1	39.4	60.1	43.1	22.4	43.7	51.1
	CE + GIoU	$39.6^{+0.2}$	$59.2^{-0.9}$	$42.9^{-0.2}$	$22.6^{+0.2}$	$43.5^{-0.2}$	$51.5^{+0.4}$
	CSE-Autoloss-A	$\mathbf{40.2^{+0.8}}$	$60.1^{+0.0}$	$43.7^{+0.6}$	$22.6^{+0.2}$	$44.3^{+0.6}$	$52.7^{+1.6}$
Cascade R-CNN R50	CE + Smooth L1	40.3	58.6	44.0	22.5	43.8	52.9
	CE + GIoU	$40.2^{-0.1}$	$58.0^{-0.6}$	$43.6^{-0.4}$	$22.4^{-0.1}$	$43.6^{-0.2}$	$52.6^{-0.3}$
	CSE-Autoloss-A	$\mathbf{40.5^{+0.2}}$	$58.8^{+0.2}$	$44.1^{+0.1}$	$22.8^{+0.3}$	$43.9^{+0.1}$	$53.3^{+0.4}$
Mask R-CNN R50	CE + Smooth L1	38.2	58.8	41.4	21.9	40.9	49.5
	CE + GIoU	$38.5^{+0.3}$	$58.8^{+0.0}$	$41.8^{+0.4}$	$21.9^{+0.0}$	$42.1^{+1.2}$	$49.7^{+0.2}$
	CSE-Autoloss-A	$\mathbf{39.1^{+0.9}}$	$59.3^{+0.5}$	$42.4^{+1.0}$	$22.4^{+0.5}$	$43.0^{+2.1}$	$51.4^{+1.9}$
FCOS R50	FL + GIoU	38.8	56.8	42.2	22.4	42.6	51.1
	CSE-Autoloss-B	$\mathbf{39.6^{+0.8}}$	$57.5^{+0.7}$	$43.1^{+0.9}$	$22.7^{+0.3}$	$43.7^{+1.1}$	$52.6^{+1.5}$
ATSS R50	FL + GIoU	40.0	57.9	43.3	23.8	43.7	51.3
	CSE-Autoloss-B	$\mathbf{40.5^{+0.5}}$	$58.3^{+0.4}$	$43.9^{+0.6}$	$23.3^{-0.5}$	$44.3^{+0.6}$	$52.5^{+1.2}$

- Across Datasets

Loss	VOC mAP (%)	BDD AP (%)
CE + L1	79.5	36.5
CE + GIoU	$79.6^{+0.1}$	$36.6^{+0.1}$
CSE-Autoloss-A	$\mathbf{80.4^{+0.9}}$	$\mathbf{37.3^{+0.8}}$

Ablation Studies

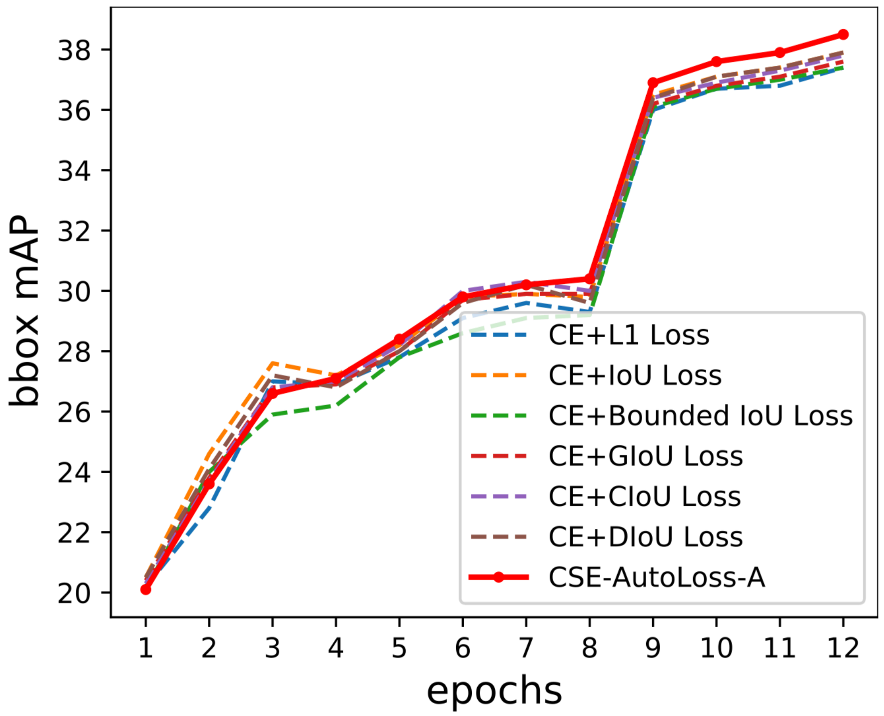
- Individual Loss Contribution

Table 6: Comparison on different loss combinations for Faster R-CNN R50 on COCO val.

Loss	AP (%)
CE + L1	37.4
CE + IoU	$37.9^{+0.5}$
CE + Bounded IoU	$37.4^{+0.0}$
CE + GIoU	$37.6^{+0.2}$
CE + DIoU	$37.9^{+0.5}$
CE + CIoU	$37.8^{+0.4}$
CE + CSE-Autoloss-A_{reg}	$37.9^{+0.5}$
CSE-Autoloss-A_{cls} + GIoU	$38.2^{+0.8}$
CSE-Autoloss-A	$\mathbf{38.5^{+1.1}}$

Table 7: Comparison on different loss combinations for FCOS R50 on COCO val.

Loss	AP (%)
FL + GIoU	38.8
FL + DIoU	$38.7^{-0.1}$
FL + CIoU	$38.8^{+0.0}$
GHM (Li et al., 2019a)	$38.6^{-0.2}$
FL + CSE-Autoloss-B_{reg}	$39.1^{+0.3}$
CSE-Autoloss-B_{cls} + GIoU	$39.4^{+0.6}$
CSE-Autoloss-B	$\mathbf{39.6^{+0.8}}$



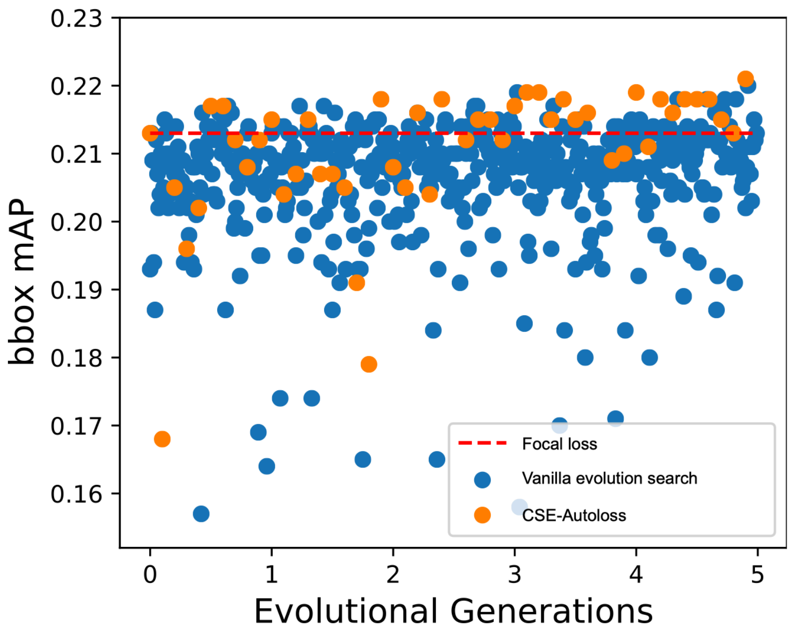
Ablation Studies

- Effectiveness of Convergence-Simulation Modules

Convergence property verification	Model optimization simulation	#Evaluated loss
		5×10^3
√		7×10^2
√	√	50

Filter out 99% loss candidates!

- Effectiveness of Proposed Search Algorithm



Search algorithm	Search branch	Another branch	#Evaluated loss	Wall-clock hours	AP (%)
Random search	Classification	GIoU	1×10^4	8×10^3	37.8
Evolution search	Classification	GIoU	5×10^3	5×10^2	38.2
CSE-Autoloss	Classification	GIoU	50	26	38.2
CSE-Autoloss	Regression	CE	15	8	37.9

CONCLUSION

Conclusion

- Propose a convergence-simulation driven evolutionary search pipeline for loss function search on object detection
- Speed up the search process by 20x by regularizing the mathematical property and optimization quality
- Validate the effectiveness of our proposed search pipeline and the search losses on various architectures and datasets

References

[1] Lin, Tsung-Yi, et al. "Focal loss for dense object detection." *Proceedings of the IEEE international conference on computer vision*. 2017.

[2] Yu, Jiahui, et al. "Unitbox: An advanced object detection network." *Proceedings of the 24th ACM international conference on Multimedia*. 2016.

[3] Rezatofighi, Hamid, et al. "Generalized intersection over union: A metric and a loss for bounding box regression." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[4] Li, Chuming, et al. "Am-lfs: Automl for loss function search." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

[5] Wang, Xiaobo, et al. "Loss function search for face recognition." *arXiv preprint arXiv:2007.06542* (2020).

[6] Li, Xiang, et al. "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection." *arXiv preprint arXiv:2006.04388* (2020).

Thank you!
Q/A?

Reporter: **Peidong Liu**
Homepage: http://perdonliu.github.io/

Paper: https://arxiv.org/abs/2102.04700
Code: https://github.com/PerdonLiu/CSE-Autoloss
Please feel free to contact me at lpd19@mails.tsinghua.edu.cn