AutoLoss-Zero: Searching Loss Functions from Scratch for Generic Tasks

Hao Li, Tianwen Fu*, Jifeng Dai, Hongsheng Li, Gao Huang, Xizhou Zhu*

haoli@link.cuhk.edu.hk, futianwen@ie.cuhk.edu.hk, {daijifeng, zhuwalter}@sensetime.com
hsli@ee.cuhk.edu.hk, gaohuang@tsinghua.edu.cn

paper code

[paper] https://arxiv.org/abs/2103.14026
[code] https://github.com/fundamentalvision/AutoLoss-Zero

CVPR JUNE 19-24 2022 NEW ORLEANS LOUISIANA

Highlights

- A general AutoML framework to search loss functions **from scratch for generic tasks** with minimal human expertise.
- Two novel techniques that bring **5000x** improved search efficiency: the Loss-Rejection Protocol and the Gradient-Equivalence-Check Strategy.
- The searched loss functions are **transferable across different models and datasets** with competitive performance.

Techniques for Improving Search Efficiency

- **Loss-Rejection Protocol**

  Directly optimize randomly initialized network predictions with the candidate loss function (instead of the network parameters), and calculate the improvement on the target metric.

$$g(L;\xi) = \frac{1}{B}\sum_{b=1}^{B} \xi\left(\hat{y}_b^*(L), y_b\right) - \xi\left(\hat{y}_b, y_b\right),$$

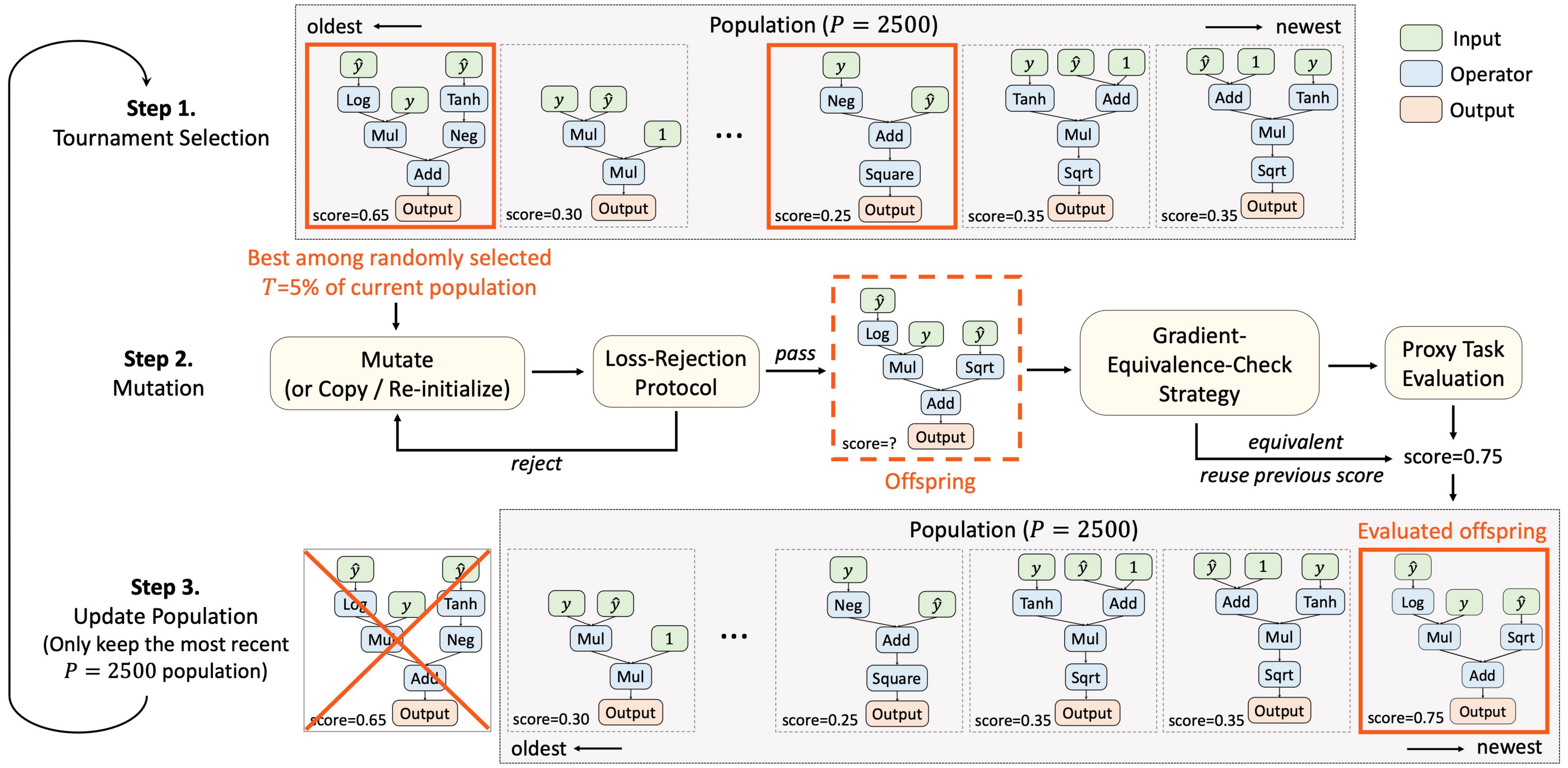
$$\text{s.t.} \quad \hat{y}_b^*(L) = \arg\min_{\hat{y}_b} \; L(\hat{y}_b, y_b),$$

- **Gradient-Equivalence-Check Strategy**

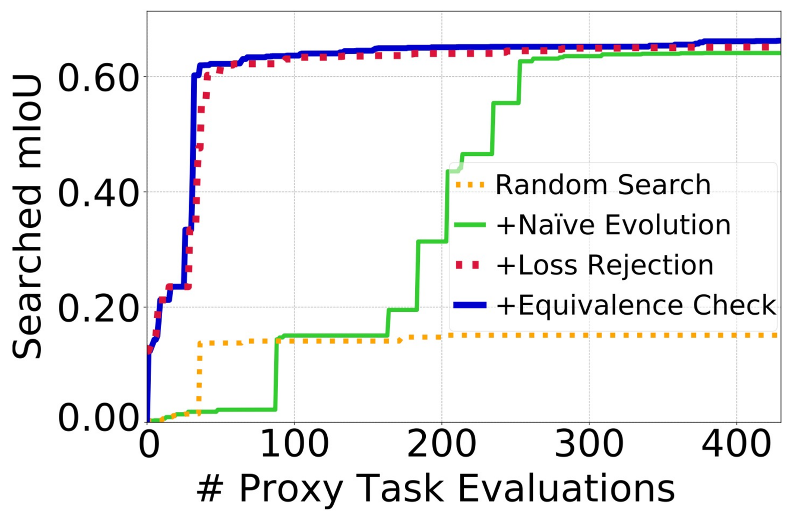
  Detect and skip the proxy task evaluation for the candidate loss functions which are equivalent to the previously evaluated loss functions. If two loss functions have the same gradient norms within two significant digits, they are considered equivalent.

$$\{\|\partial L/\partial \hat{y}_b\|_2\}_{b=1}^{B}$$

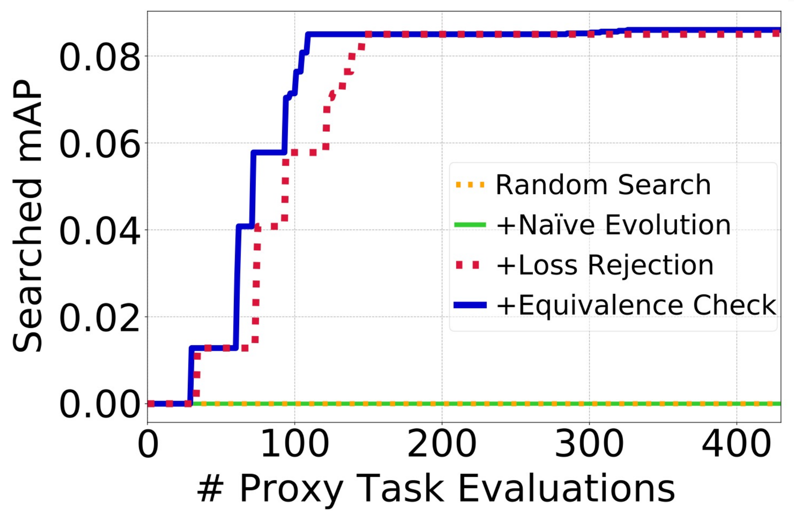
Pipeline of the Evolutionary Search



Ablations on Search Efficiency



(a) Semantic Segmentation



(b) Object Detection

	Speed-Up	# Explored Losses
Naïve Evolution	$1\times$	$\sim$300
+ Loss-Rejection Protocol	$\sim$700$\times$	$\sim 2.1\times10^5$
+ Gradient-Equivalence-Check Strategy	$\sim$1000$\times$	$\sim 3.2\times10^5$
+ †Stop Training for Invalid Loss Values	$\sim$5000$\times$	$\sim 1.5\times10^6$

- **Comparison against Random Search**

Loss Function	mIoU
Random Search	2.2
Ours	80.7

(a) Semantic Segmentation

Loss Function	mAP
Random Search	0.0
Ours	38.1

(b) Object Detection

Experiments

- **Semantic segmentation results on PASCAL VOC**

Loss Function		FWIoU	gAcc	mAcc	BIoU	mIoU	BF1
Cross Entropy		91.3	**95.2**	87.3	70.6	78.7	65.3
WCE [53]		85.6	91.1	**92.6**	61.8	69.6	37.6
DiceLoss [38]		91.3	95.1	87.5	69.9	77.8	64.4
Lovàsz [2]		**91.8**	**95.4**	88.6	72.5	79.7	66.7
DPCE [4]		**91.8**	**95.5**	87.8	71.9	79.8	66.5
SSIM [44]		**91.7**	**95.4**	87.9	71.5	79.3	66.4
FWIoU	ASL [31]	**91.9**	**95.4**	89.2	75.1	80.0	65.7
	Ours	**91.7**	**95.2**	87.7	72.9	78.7	64.6
gAcc	ASL [31]	**91.8**	**95.5**	89.0	74.1	79.7	64.4
	Ours	**91.7**	**95.3**	88.7	73.6	79.4	64.8
mAcc	ASL [31]	85.9	91.3	**92.7**	72.9	69.8	35.6
	Ours	89.2	93.7	**92.6**	73.7	75.3	44.1
BIoU	ASL [31]	69.9	62.6	81.3	**79.2**	49.0	39.0
	Ours	69.5	80.5	67.1	**79.3**	50.0	34.4
mIoU	CSE [37]	91.4	95.2	87.0	72.6	78.1	64.1
	CSE-RandInit	89.6	93.9	83.1	64.6	71.9	56.5
	AML [49]	59.5	64.4	4.9	1.3	4.0	0.4
	ASL [31]	**92.1**	**95.7**	88.2	73.4	**81.0**	68.9
	Ours	**92.1**	**95.7**	89.1	74.1	**80.7**	66.0
BF1	CSE [37]	91.8	95.4	88.5	73.7	79.4	65.1
	CSE-RandInit	69.3	75.6	9.0	3.0	5.3	1.0
	AML [49]	0.5	2.6	4.7	1.7	0.8	1.1
	ASL [31]	1.0	2.7	6.5	7.4	1.9	74.8
	Ours	4.2	9.1	11.9	26.1	7.3	**76.7**

- **COCO detection**

Loss Function				mAP
Cls_{RPN}	Reg_{RPN}	Cls_{RCNN}	Reg_{RCNN}	
CE	L1	CE	L1	37.3
CE	L1	CE	IoULoss [63]	37.9
CE	L1	CE	GIoULoss [52]	37.6
CE	L1	CSE-Auto-A [37]		38.5
CE	L1	CSE-RandInit		0.0
CE	L1	Ours		38.0
Ours				38.1

- **Generalization on different models and datasets**

Dataset		Cityscapes		VOC			
Network		R101-DLv3+		R50-DLv3+		R101-PSP	
Loss Function		mIoU	BF1	mIoU	BF1	mIoU	BF1
Cross Entropy		80.0	62.2	76.2	61.8	77.9	64.7
mIoU	ASL [31]	**80.7**	66.5	**78.4**	66.9	**78.9**	65.7
	Ours	**80.4**	63.8	**78.0**	62.8	**78.5**	64.9
BF1	ASL [31]	6.7	**78.0**	1.4	70.8	1.6	71.8
	Ours	16.0	**77.5**	10.4	**79.2**	11.5	**76.4**

- **COCO pose estimation**

Loss Function	mAP
MSE	71.5
Ours	72.0

- **COCO instance segmentation**

Loss Function	mAP
CE + L1 + CE + L1 + CE	34.6
CE + L1 + CE + IoULoss [63] + CE	34.4
CE + L1 + CE + GIoULoss [52] + CE	34.7
Ours	34.8