## A. Implementation Details

For the evolutionary algorithm, the population is initialized with $K = 20$ randomly generated loss functions, and is restricted to most recent $P = 2500$ loss functions. The ratio of tournament selection [20] is set as $T = 5\%$ of current population. During random initialization and mutations, the sampling probabilities for all the operators in the primitive operator set $\mathcal{H}$ are the same. The initial depth of computational graphs is $D = 3$. For the loss-rejection protocol and the gradient-equivalence-check strategy, $B = 5$ samples are randomly selected from the training set $\mathcal{S}_{train}$.

In the proposed loss-rejection protocol, the threshold for unpromising loss function is set as $\eta = 0.6$. We use stochastic gradient descent with momentum to optimize Eq. (3) for 500 iterations. The learning rate for the loss-rejection protocol is 0.001. The momentum factor is set as 0.9. There is no weight decay. Since the predictions of different samples and spatial positions are optimized independently in Eq. (3), we do not normalize along any dimensions when aggregating the output tensor to the final loss value in the loss-rejection protocol.

We stop the search when the total number of proxy task evaluations reaches 500. The best loss function is then used for re-training experiments. All the experiments are conducted on 4 NVIDIA V100 GPUs.

### A.1. Semantic Segmentation

**Datasets.** PASCAL VOC 2012 [17] with extra annotations [21] is utilized for our experiments. During search, we randomly sample 1500 training images in PASCAL VOC to form the evaluation set $\mathcal{S}_{\text{eval}}$, and use the remaining training images as the training set $\mathcal{S}_{\text{train}}$. The target evaluation metrics include Mean IoU (mIoU), Frequency Weighted IoU (FWIoU), Global Accuracy (gAcc), Mean Accuracy (mAcc), Boundary IoU (BIoU) [28] and Boundary F1 Score (BF1) [14]. The first four metrics measure the overall segmentation accuracy, and the other two metrics evaluate the boundary accuracy.

**Implementation Details.** During search, we use DeepLabv3+ [8] with ResNet-50 [24] as the network. The softmax probabilities $y$ and the one-hot ground-truth labels $\hat{y}$ are used as the inputs of loss functions, both of which are of shape $(N, C, H, W)$. For the boundary metrics, we use the pre-computed boundaries of $y$ as the training targets. Following [31], we simplify the proxy task by downsampling the input images to the resolution of $128 \times 128$, and reducing the training schedule to 3 epochs (1/10 of the normal training schedule) with a mini-batch size of 32. We use stochastic gradient decent with momentum to train the network. The initial learning rate is 0.02, which is decayed by polynomial with power 0.9 and minimum learning rate $10^{-4}$. The momentum and weight decay factors are set to 0.9 and $5 \times 10^{-4}$, respectively. For faster convergence, the learning rate of the segmentation head is multiplied by 10. After the search procedure, we re-train the segmentation networks with ResNet-101 as the backbone for 30 epochs. The input image resolution is $512 \times 512$. The re-training setting is the same as [8], except that the searched loss function is utilized.

### A.2. Object Detection

**Datasets.** We conduct experiments on large-scale object detection dataset COCO [34]. In the search experiments, we randomly sample 5000 images from the training set for validation purpose, and sample $1/4$ of the remaining images for network training. The target evaluation metric is Mean Average Precision (mAP).

**Implementation Details.** We use Faster R-CNN [51] with ResNet-50 [24] and FPN [32] as the detection network. There are 4 loss branches, *i.e.*, the classification and regression branches for the RPN [51] sub-network and Fast R-CNN [19] sub-network. We search for loss functions of the 4 branches simultaneously from scratch. The inputs of loss functions for the classification branches are the softmax probabilities and the one-hot ground-truth labels. Following [37], we use the intersection, union and enclosing areas between the predicted and ground-truth boxes as the regression loss inputs. The loss weights are set to 1.0 for classification branches and 10.0 for regression branches, which follows [52].

During search, the initialization / mutation process is repeated for each loss branch separately until it passes the loss-rejection protocol. In the loss-rejection process of each loss branch, the predictions of the other branches are set as the ground-truth targets. For the RPN sub-network, the correlation score $g(L; \xi)$ is calculated with the predicted region proposals and the corresponding training targets.

We train the network with $1/4$ of the COCO data for 1 epoch as the proxy task. We further simplify the network by only using the last three feature levels of FPN, and reducing the channels of the detection head by half. The learning rate is set to 0.04 with a batch size of 32, and a linear learning rate warm-up of 250 iterations is applied. After the search procedure, we re-train the detection network with the searched loss functions for 12 epochs. The re-training hyper-parameters are the same as the default settings of MMDetection [6].

### A.3. Instance Segmentation

**Datasets.** We conduct experiments on COCO [34]. In the search experiments, we randomly sample 5000 images from the training set for validation purpose, and sample $1/4$ of the remaining images for network training. The target metric is mAP with IoU defined on masks.

| **Task** | **Metric** | **Formula** |
|---|---|---|
| Seg | mIoU | $-\frac{1}{2}\log\left(\tanh\left(\frac{y}{1+y}\sqrt{\hat{y}}\right)(\hat{y}^2+y)\right)$ |
| | FWIoU | $\log(y\log(y+(y+\hat{y})/\hat{y}))$ |
| | gAcc | $\exp\left(\left(y-\sqrt{\hat{y}}\right)^2\right)$ |
| | mAcc | $\tanh\left(\sqrt{\text{Mean}_{nhw}\left(-\log(\hat{y})\sqrt{\text{Max-Pooling}\left(\sqrt{2y}\right)}\right)}\right)$ |
| | BIoU | $-\log\left(\hat{y}^3y\sqrt{-\text{Min-Pooling}(-\hat{y})}\exp\left((\hat{y}\exp(\hat{y}))^2\right)\right)$ |
| | BF1 | $\exp\left(\text{Mean}_{nhw}\left(-\text{Max-Pooling}(y)\log\left(\text{Max-Pooling}(\hat{y})\right)\right)+\tanh\left(\text{Mean}_{nhw}\left(2\hat{y}+1\right)\right)\right)$ |
| Det | mAP | Cls$_{\text{RPN}}$ $\exp(\tanh(y))/\hat{y}$ |
| | | Reg$_{\text{RPN}}$ $e/(i+u)$ |
| | | Cls$_{\text{RCNN}}$ $-y\log(\hat{y})$ |
| | | Reg$_{\text{RCNN}}$ $-\log(i/e)$ |
| Ins | mAP | Cls$_{\text{RPN}}$ $-y\hat{y}/\tanh(\hat{y})$ |
| | | Reg$_{\text{RPN}}$ $e/(i+u)$ |
| | | Cls$_{\text{RCNN}}$ $-y\log(\hat{y})$ |
| | | Reg$_{\text{RCNN}}$ $-\sqrt{i}/\sqrt{e}$ |
| | | Mask $\lvert\log(y\hat{y})\rvert$ |
| Pose | mAP | $(\hat{y}-16y^4)^2+\hat{y}^4$ |

Table 9. Discovered Loss Functions. "Seg", "Det", "Ins", and "Pose" denotes the tasks of semantic segmentation, object detection, instance segmentation, and pose estimation, respectively. Max-Pooling and Min-Pooling have kernel size $3\times3$. A small positive number $\epsilon = 10^{-12}$ is added in $\log(\cdot)$, $\sqrt{\cdot}$, and the denominators of division operations to avoid infinite values or gradients. $\hat{y}$ and $y$ denote the network prediction and the training target of the corresponding branch. $i$, $u$, and $e$ in the box regression loss branches refer to the intersection, union, and enclosed areas between the predicted and the target bounding boxes, respectively, which follows [37].

**Implementation Details.** Mask R-CNN [23] with ResNet-50 [24] and FPN [32] is used as the network. We conduct the search for all the 5 loss branches simultaneously. The inputs of loss functions for the classification and regression branches are the same as in object detection. The prediction used as loss inputs for each RoI of the mask branch is the per-pixel softmax probabilities indicating foreground or background.

The proxy task is the same as for object detection. After the search procedure, we re-train the detection network with the searched loss functions for 12 epochs. We use the default hyper-parameters of MMDetection [6] for re-training the networks with our searched loss functions.

## A.4. Pose Estimation

**Datasets.** Experiments are conducted on COCO [34], which contains 250,000 person instances labeled with 17 keypoints each. During search, we randomly select 5000 image from the training set for proxy task evaluation. The target evaluation metric is keypoint mAP [62], which is very similar to the mAP in object detection, where object keypoint similarity (OKS) is used to substitute the bounding box IoU.

**Implementation Details.** We use [62] with Resnet-50 [24] as the network. Following the practice in [12], person detection results provided by [62] are utilized. For each joint of each detected box in the person detection results, the network predicts a $64 \times 48$ heatmap, and the target heatmap is constructed as in [62]. Loss function is applied only on visible joints.

During search, we train the network using the Adam [27] optimizer for 4 epochs as the proxy task. The learning rate is $2.5 \times 10^{-4}$ with a cosine annealing schedule and a linear warm-up for 250 iterations. The batch size is set as 256. After the search procedure, we re-train the network with the searched loss functions for 210 epochs using the default training settings of MMPose [12].

## B. Formulas of the Discovered Loss Functions

Table 9 shows the formulas of the discovered loss functions. The formulas are simplified manually by removing useless operators (*e.g.* squaring the constant one or taking the absolute value of a non-negative quantity).

## C. Search Efficiency

We ablate the search efficiency of AutoLoss-Zero on semantic segmentation with the mIoU metric and object detection with the mAP metric. Starting from random search, we add each component of AutoLoss-Zero sequentially to verify its effectiveness. Figure 4 shows the results. The proposed loss-rejection protocol greatly improves the search efficiency for both tasks, and the gradient-equivalence-check strategy is also helpful. For semantic segmentation which only contains a single loss branch, naïve evolution can also discover the proper losses despite its inefficiency. For object detection with 4 loss branches, the exponentially increased sparsity of the search space brings great difficulty to the search. Therefore, without the loss-rejection protocol, no loss functions with scores greater than zero can be discovered within a reasonable time. Table 8 further presents the number of loss functions that can be explored in 48 hours by AutoLoss-Zero. Over $10^6$ loss functions can be explored, which ensures that AutoLoss-Zero can explore the huge and sparse search space within a reasonable time.

## D. Licences of Assets

### D.1. Datasets

**Pascal VOC** [17] uses images from Flickr, which is subject to the Flickr terms of use [18].

**Cityscapes** [13] is licenced under the Terms of Use of Cityscapes [9].

**COCO** [34]. The annotations are under the Creative Commons Attribution 4.0 License [10]. The images are subject to the Flickr terms of use [18].

### D.2. Code

**MMSegmentation** [11], **MMDetection** [6], **MM-Pose** [12] are licenced under the Apache License 2.0, and the copyright holder is Open-MMLab.