Supplement Materials for Bridging Cross-task Protocol Inconsistency for Distillation in Dense Object Detection

In the supplementary material, we provide the following experimental results and details:

- Section A: Algorithm to calculate the distillation loss.
- Section B: Implementation details.
- Section C: More experimental results about self-KD, heterogeneous backbones, base detectors, Pascal VOC dataset, inheriting initialization and response-based distillation.
- Section D: Visualization about intermediate training phases and predicted results.

A. Algorithm

In this section, we propose an algorithm to calculate the distillation loss, which is illustrated in Algorithm A1.

Algorithm A1 The algorithm to calculate the distillation loss.

Require: Training data $x_{i\{i=1,\cdots,n\}}$, student dense object detector S_{det}, parameters θ_s of S_{det}, student dense object detector T_{det}, parameters θ_t of T_{det}, positions pos of anchors

Uniformly sample a minibatch of training data $B^{(t)}$
for $x_i \in B^{(t)}$ **do**
  l^s, o^s = $S_{det}(x_i;\theta_s)$
  l^t, o^t = $T_{det}(x_i;\theta_t)$
  procedure CLASSIFICATION(l^s, l^t)
    $p^{s\prime}$ = Sigmoid(l^s)
    $p^{t\prime}$ = Sigmoid(l^t)
    $w = \left|p^{t\prime} - p^{s\prime}\right|$
    $\mathcal{L}_{cls}^{dis}(x_i) = \sum w \cdot BCE(l^{s\prime}, l^{t\prime})$
  end procedure
  procedure LOCALIZATION(o^s, o^t)
    b^s = Decoder(pos, o^s)
    b^t = Decoder(pos, o^t)
    u' = IoU(b^s, b^t)
    $\mathcal{L}_{loc}^{dis}(x_i) = \sum_{i=1}^{n} max(w_{\cdot,j}) \cdot (1 - u_i')$
  end procedure
end for

B. Implementation Details

B.1. Main Experiment

Our implementation is based on Pytorch and mmdetection [2]. Different training schedules are set to ensure fair comparison with previous methods, such as 1x (namely 12 epochs) and 2x (namely 24 epochs). We use SGD optimizer with momentum and weight decay set to 0.9 and 0.0001, respectively. The initial learning rate is set to 0.01. Our proposed method employs α_1 and α_2 to balance the classification and localization distillation losses, which are set to 1.0 and 4.0, respectively. All experiments are conducted on 8 RTX 3090 GPUs, with a batch size of 2 images per GPU.

B.2. Combined with Feature-based Methods

Our implementation is based on Pytorch and mmrazor [3]. We adopt the same training schedules, SGD optimizer, and learning rate settings as described in Section B.1. The hyperparameters α_1 and α_2 are set to 0.25 and 2.0, respectively. All experiments are conducted on 8 RTX 3090 GPUs with 2 images per GPU.

C. More Experiment

C.1. Self KD

We have demonstrated the effectiveness of our proposed approach in transferring knowledge from a powerful teacher to a compact student. Then, in cases where a stronger teacher model is not available, self-KD [5, 8] has emerged as a popular technique for classification. In the context of dense object detection, we simulate similar scenarios by setting $S_{det} = T_{det}$, where S_{det} and T_{det} denote the student and teacher detectors, respectively. Our approach also improves the performance under the self-KD strategy with lightweight detectors, as shown in Table A1. In contrast, LD [10] leads to performance degradation in these scenarios.

Method	Schedule	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
GFocal-Res34(Student)	1x	38.9	56.6	42.2	21.5	42.8	51.4
LD [10] self-KD	1x	38.6	56.0	41.7	21.0	42.4	50.4
Ours self-KD	1x	**39.4**	**57.2**	**42.6**	**21.7**	**43.4**	**51.6**
GFocal-Res18(Student)	1x	35.8	53.1	38.2	18.9	38.9	47.9
LD [10] self-KD	1x	35.0	52.1	37.7	18.6	38.6	46.0
Ours Self-KD	1x	**36.2**	**53.5**	**38.9**	**19.3**	**39.6**	**48.3**

Table A1. Quantitative evaluation results of our proposed method and other logits-based distillation techniques for self-KD scenario on MS COCO *val2017*.

C.2. Heterogeneous Backbone

Recently, powerful backbones such as Swin-Transformer have exhibited remarkable performance in various computer vision tasks. Nonetheless, CNN-based dense object detectors remain extensively employed in practical applications due to their high speed and ease of deployment. Due to the large gap in feature representations between the two architectures, applying feature-based methods from Transformer-based detectors to CNN-based detectors is challenging. To address this issue, we propose a prediction-level distillation method that is feature-free and particularly suitable for this task. As a result, we can use more powerful teacher detectors to enhance the performance of compact student detectors. As demonstrated in Table A2, our proposed method is effective when transferring knowledge between detectors with heterogeneous backbones.

Method	Schedule	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
GFocal-SwinT(Teacher)	2x	47.3	66.2	51.4	31.8	50.9	60.7
GFocal-Res50(Student)	1x	40.1	58.2	43.1	23.3	44.4	52.5
Ours	1x	43.0	61.5	46.7	25.7	47.3	55.9
GFocal-ResX101DCN(Teacher)	2x	48.1	67.1	52.5	29.7	52.1	62.7
GFocal-Res50(Student)	1x	40.1	58.2	43.1	23.3	44.4	52.5
Ours	1x	42.6	61.4	46.4	26.1	46.4	55.1
GFocal-Res50(Teacher)	2x	42.9	61.2	46.5	27.3	46.9	53.3
GFocal-Res50(Student)	1x	40.1	58.2	43.1	23.3	44.4	52.5
Ours	1x	42.8	61.2	46.4	26.0	47.0	54.1
GFocal-Res101(Teacher)	2x	44.9	63.1	49.0	28.0	49.1	57.2
GFocal-MobileNetv2(Student)	1x	32.6	48.5	34.9	18.0	34.6	43.5
Ours	1x	35.1	51.8	37.8	19.1	37.9	45.6

Table A2. Quantitative evaluation results of proposed distillation method for heterogeneous backbones on MS COCO *val2017*.

C.3. Base Detector

In this subsection, we evaluate our proposed method on additional base detectors, such as ATSS [9] and YOLOX. The results in Table A6 indicate that our method achieves comparable gains to the state-of-the-art feature-based methods. The

significant improvement in detector mAP on ATSS [9] and YOLOX further confirms the robust generalization ability of our proposed method.

Method	Schedule	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ATSS-Res101(Teacher)	1x	41.5	59.9	45.2	24.2	45.9	53.3
ATSS-Res50(Student)	1x	39.4	57.6	42.8	23.6	42.9	50.3
Ours	1x	**41.4**	**59.9**	**45.1**	**25.1**	45.6	53.5
PKD	1x	41.3	59.2	44.6	24.1	45.6	53.9
PKD + Ours	1x	**41.4**	59.5	44.8	23.7	**45.7**	**54.1**
YOLOX-s (Teacher)	1x	40.3	59.1	43.4	23.5	44.5	53.1
YOLOX-tiny (Student)	1x	31.8	49.0	33.8	12.3	34.9	47.8
Ours	1x	34.2(+2.4)	**52.0(+3.0)**	**36.4(+2.6)**	**14.7(+2.4)**	39.1(+4.2)	50.0(+2.2)

Table A3. Quantitative evaluation results of different distillation methods for ATSS on MS COCO *val2017*.

C.4. Pascal VOC Dataset

Many recent object detection distillation methods, e.g., [7, 1], only report experimental results on COCO. We follow their experimental setups. Additionally, we expand our evaluations on Pascal VOC. Table A4 shows that our method increases the performance from 52.2 to 55.2.

C.5. Inheriting Initialization

As shown in Table A5, equipping with the inheriting strategy, the performance of our method further increases by 0.5 mAP.

C.6. Response-based Distillation

GID [4] proposes response-based distillation. Our method is different with GID [4]. On the one hand, the response-based distillation in GID is motivated by "the definition of outputs from the detector head varies from model to model". In contrast, our method is motivated by an in-depth analysis of the main challenge faced by logit-based distillation techniques in object detection. We identify the cross-task protocol inconsistency between distillation and classification and propose cross-task consistent protocols as a solution. This finding offers an interesting insight to the research community. On the other hand, GID selects part regions of images (i.e., GIs) for distillation. Our method incorporates all regions of the image in distillation, and we use score-aware weighting for different regions, eliminating the need for complex GI designs. We compare our method with GID in Table A6. The results in Table A6 indicate the superiority of our method over GID.

Method	mAP	AP_{50}	AP_{75}
GFocal-Res50 (Teacher)	56.4	79.1	61.3
GFocal-Res18 (Student)	52.2	75.8	56.4
Ours	55.2	78.1	59.2

Table A4. Quantitative evaluation results on Pascal VOC.

Method	mAP	AP_{50}	AP_{75}
FCOS-Res101 (Teacher)	40.8	60.0	44.0
FCOS-Res50 (Student)	36.6	56.0	38.8
PKD + Ours w/o inheriting	40.2	59.5	43.0
PKD + Ours w/ inheriting	40.7	60.0	43.5

Table A5. Quantitative evaluation results with inherit strategy.

Method	Schedule	mAP	AP_{50}	AP_{75}
Retina-Res101 (Teacher)	1x	38.1	58.3	40.9
Retina-Res50 (Student)	1x	36.2	55.8	38.8
Response-based Distillation [4]	1x	37.9	57.8	41.1
Ours	1x	**39.2**	**59.1**	**42.4**

Table A6. Quantitative evaluation results of our method and Response-based Distillation in GID.

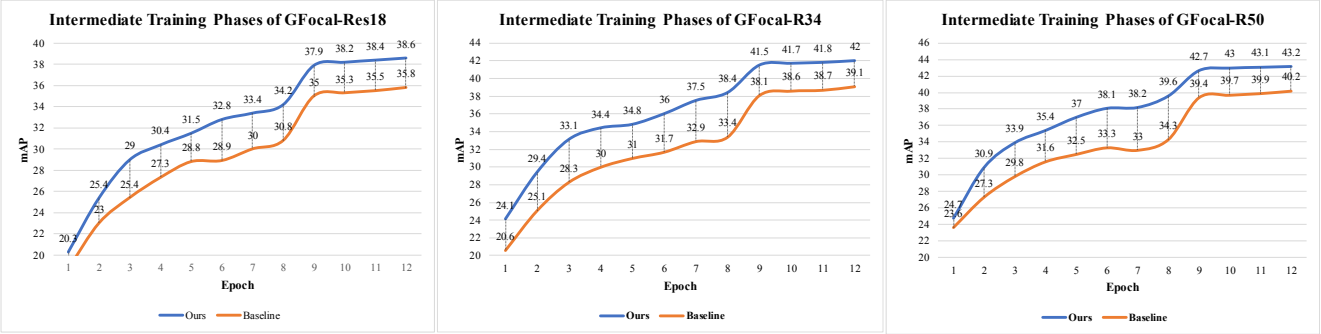

Figure A1. Visualization of intermediate training phases on GFocal-Res18, GFocal-R34 and GFocal-R50.

D.2. Prediction Visualization

We provide visual evidence to validate the effectiveness of our proposed method by presenting images from MS COCO [6] *val2017* with various score thresholds. Figure A2 depicts that our method outperforms LD [10] in detecting more high-quality bounding boxes, particularly under high score thresholds. This result implies that our proposed method offers a more favorable classification score distribution.

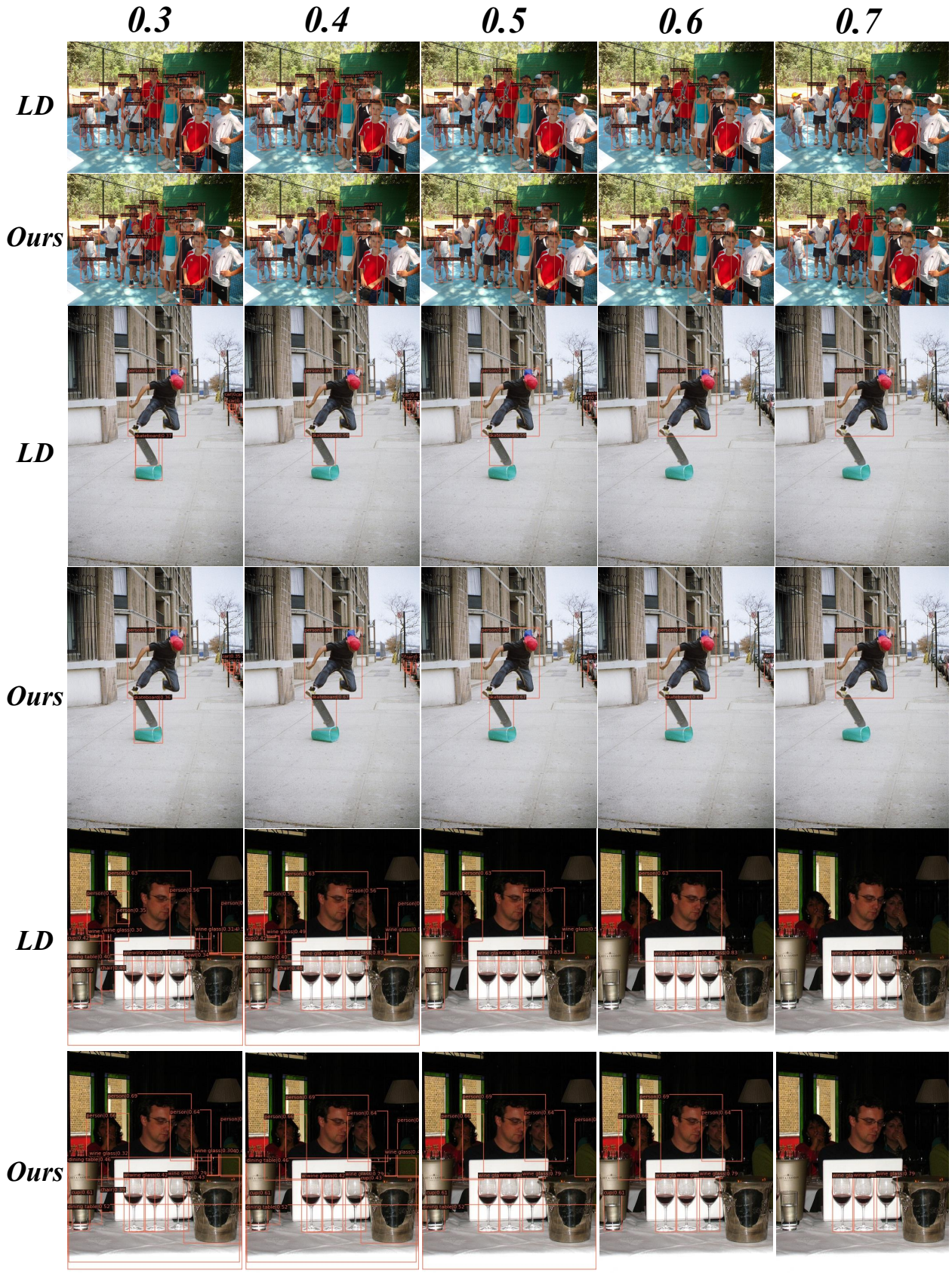

Figure A2. Visualization compared to LD [10]. *0.3* means under score threshold = 0.3. Best viewed in color.

References

[1] Weihan Cao, Yifan Zhang, Jianfei Gao, Anda Cheng, Ke Cheng, and Jian Cheng. Pkd: General distillation framework for object detectors via pearson correlation coefficient. *arXiv preprint arXiv:2207.02039*, 2022. 3

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[3] MMRazor Contributors. Openmmlab model compression toolbox and benchmark. https://github.com/open-mmlab/mmrazor, 2021. 1

[4] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. In *Proc. CVPR*, pages 7842–7851, 2021. 3, 4

[5] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proc. ICML*, pages 1607–1616. PMLR, 2018. 2

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, pages 740–755. Springer, 2014. 4

[7] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. In *Proc. ECCV*, pages 53–69. Springer, 2022. 3

[8] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proc. ICCV*, pages 3713–3722, 2019. 2

[9] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proc. CVPR*, pages 9759–9768, 2020. 2, 3

[10] Zhaohui Zheng, Rongguang Ye, Ping Wang, Dongwei Ren, Wangmeng Zuo, Qibin Hou, and Ming-Ming Cheng. Localization distillation for dense object detection. In *Proc. CVPR*, pages 9407–9416, 2022. 2, 4, 5