Relational Surrogate Loss Learning

ICLR 2022

Tao Huang[1,2] Zekang Li[3] Hua Lu[4] Yong Shan[3] Shusheng Yang[4]
Yang Feng[3] Fei Wang[5] Shan You[2] Chang Xu[1]

[1]The University of Sydney [2]SenseTime Research [3]Chinese Academy of Sciences
[4]Huazhong University of Science and Technology [5]University of Science and Technology of China

Loss Functions in Machine Learning
Problem Statement

Evaluation metrics are used to measure the performance of models

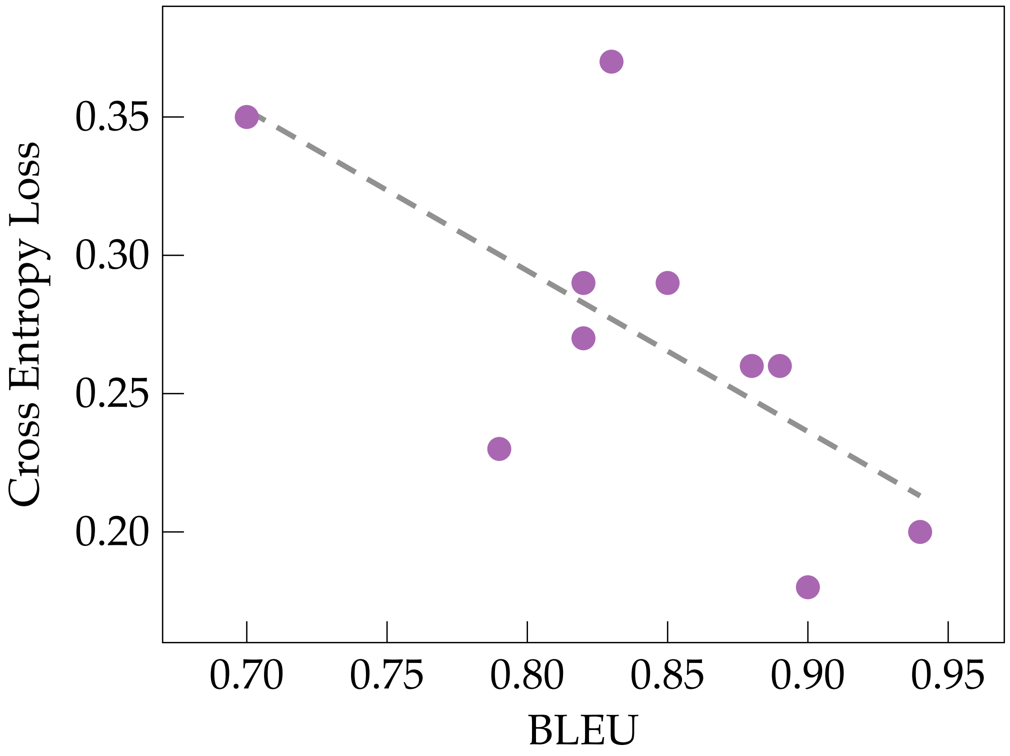
but

Most of them are non-differentiable and non-decomposable

Loss functions are designed as proxies of evaluation metrics

but

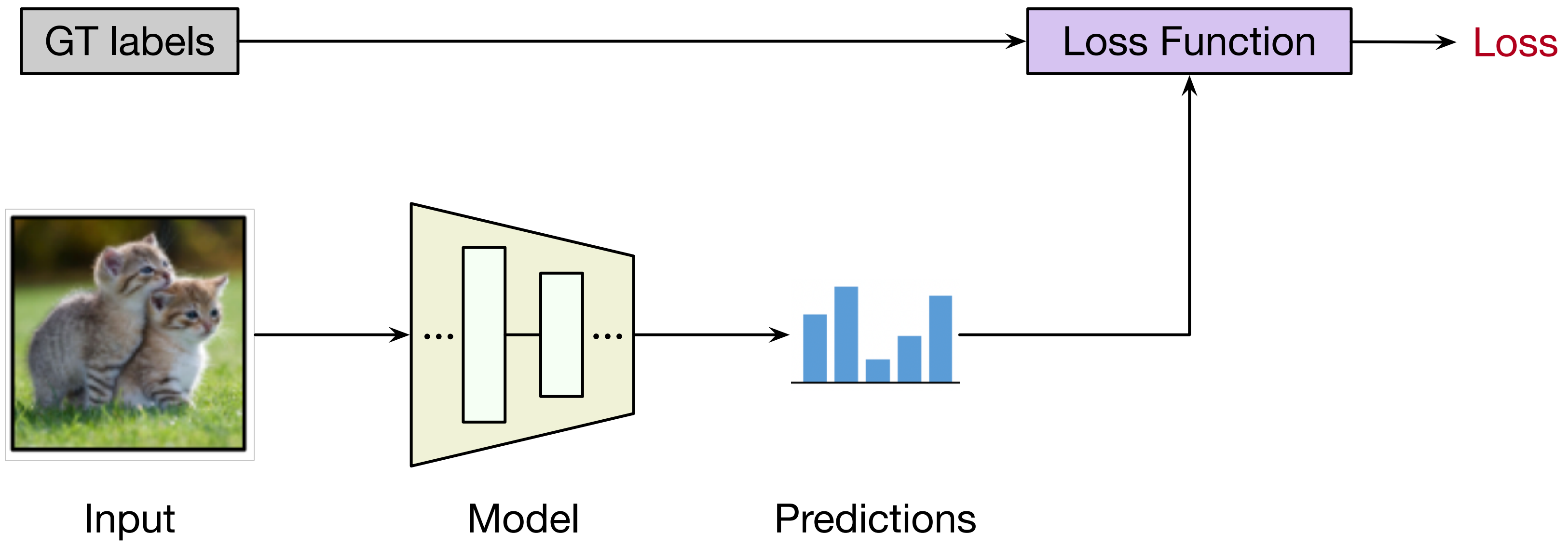
Manually designing a loss
- Requires **expertise** on specific tasks
- **Hard** to align well with the metric



BLEU vs. CE loss of samples on neural machine translation task, showing a weak correlation (Spearman: -0.58).

How to design loss functions automatically?

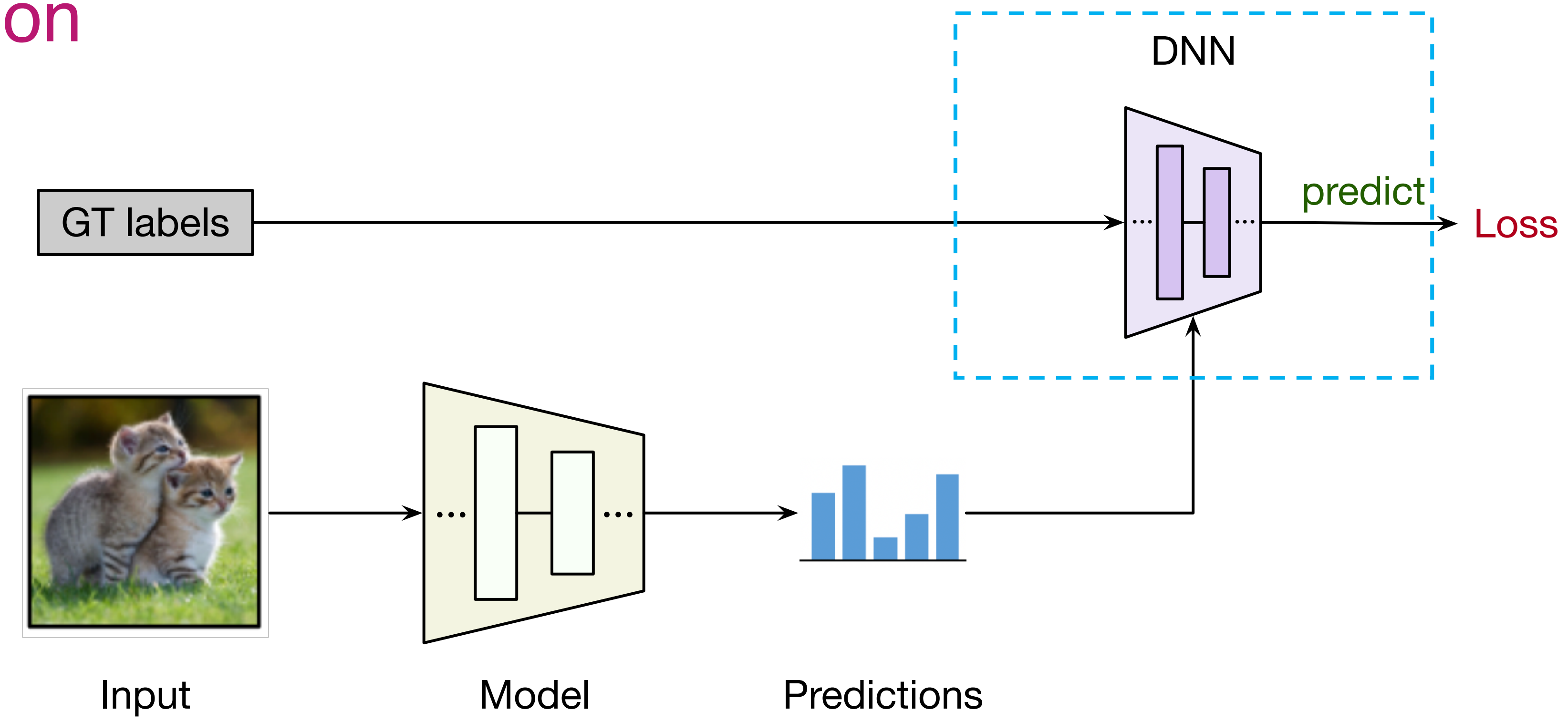
Surrogate Loss Learning
Introduction



Training models with conventional loss function.

Surrogate Loss Learning

Introduction

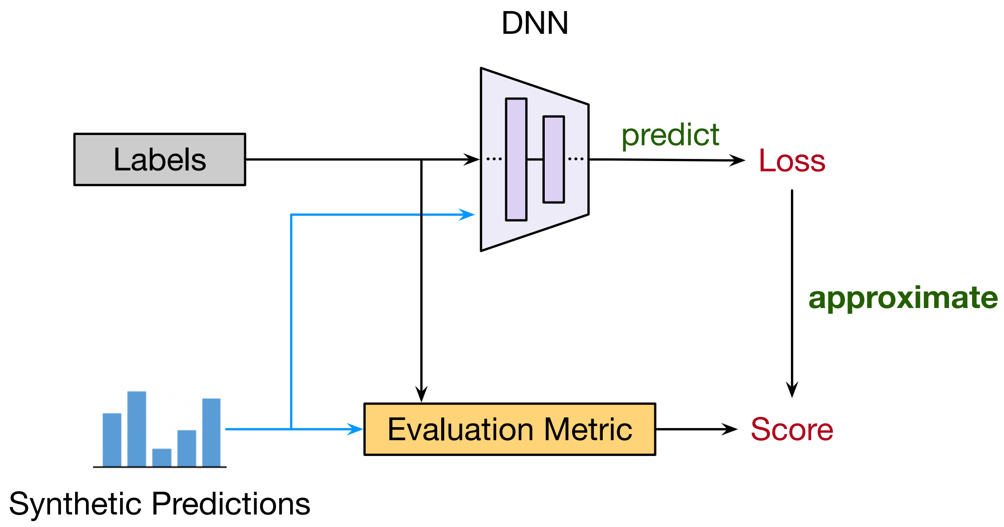


Training models with surrogate loss.

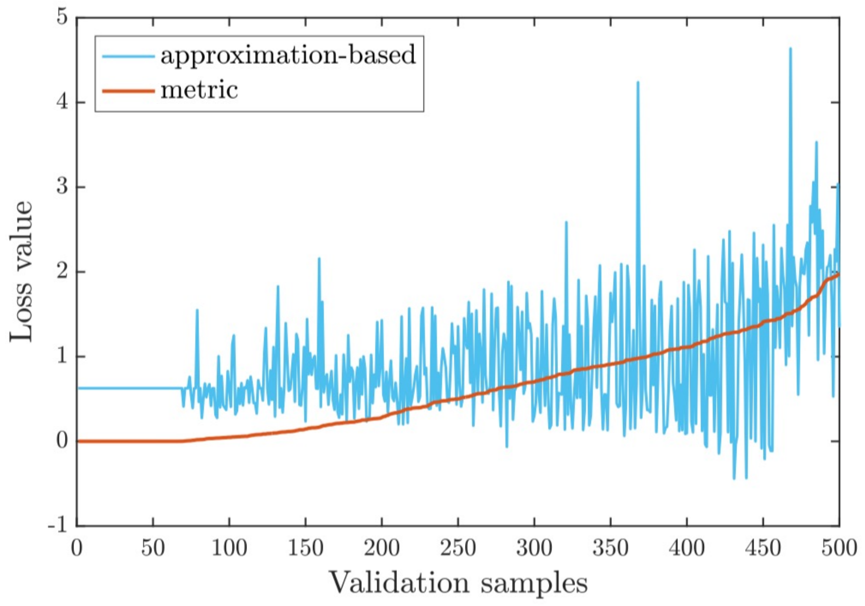
- Approximate the evaluation metrics using a deep neural network (DNN)
- Replace the conventional loss function with the learned DNN

Surrogate Loss Learning

Limitations



Training surrogate loss.



Predictions of evaluation metric and learned surrogate loss. The loss values wave drastically.

Poor performance

- ▶ The surrogate loss cannot **fully recover** the metric values

Weak generalizability

- ▶ Easy to overfit on the training samples

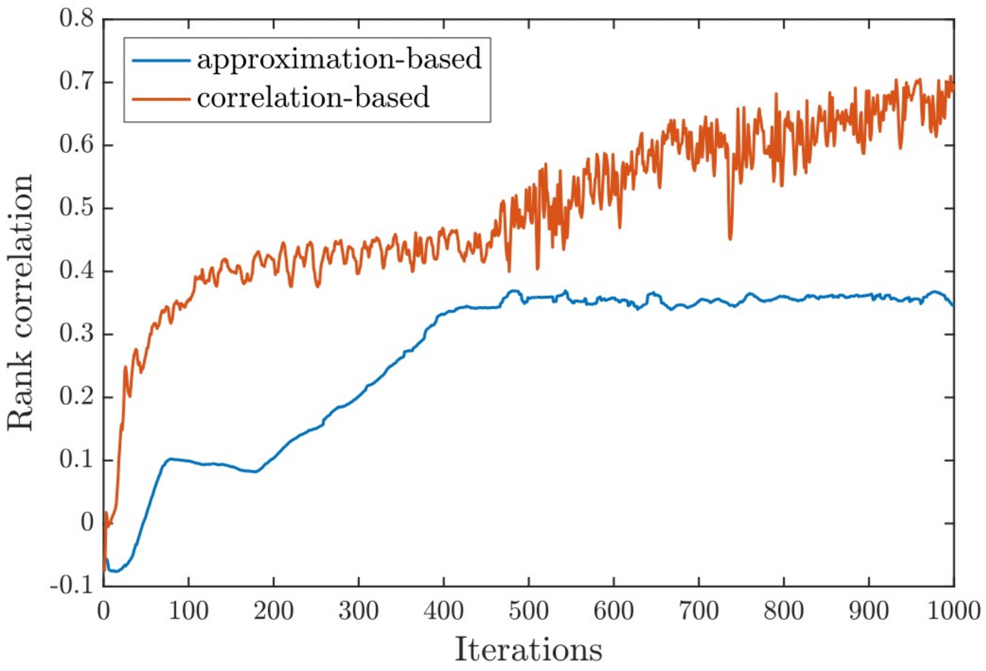
  → need to train surrogate loss and model **alternatively**
- ▶ The learned surrogate loss cannot generalize to different models and tasks

Relational Surrogate Loss Learning
Motivation

- Evaluation metrics (losses) are used to distinguish whether a model is better or worse than another
  → we only need to keep the **relative rankings** of samples between the loss and metric

Our solution:

- Only learn **rank correlations** between the loss and metric instead of **approximating** the metric
- Propose a differentiable rank correlation loss using differentiable sorting algorithm (Petersen et al., 2021)



Rank correlations between loss and metric (approximation-based loss vs. our correlation-based loss).

[1] Petersen, Felix, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Differentiable sorting networks for scalable sorting and ranking supervision. In International Conference on Machine Learning, pp. 8546-8555. PMLR, 2021.

Relational Surrogate Loss Learning
Gradient Penalty

We only constrain the correlation in the training of surrogate loss
→ its first-order derivative changes drastically
Solution: an additional gradient penalty term to enforce the Lipschitz constraint

$$\mathcal{L}_{\text{penalty}} = (\| \nabla_y \mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}; \boldsymbol{\theta}_l) \|_2 - 1)^2.$$

Loss term of gradient penalty (Eq.(8) in the paper).

Effect of gradient penalty:

- Stabilize model training
- Improve generalizability

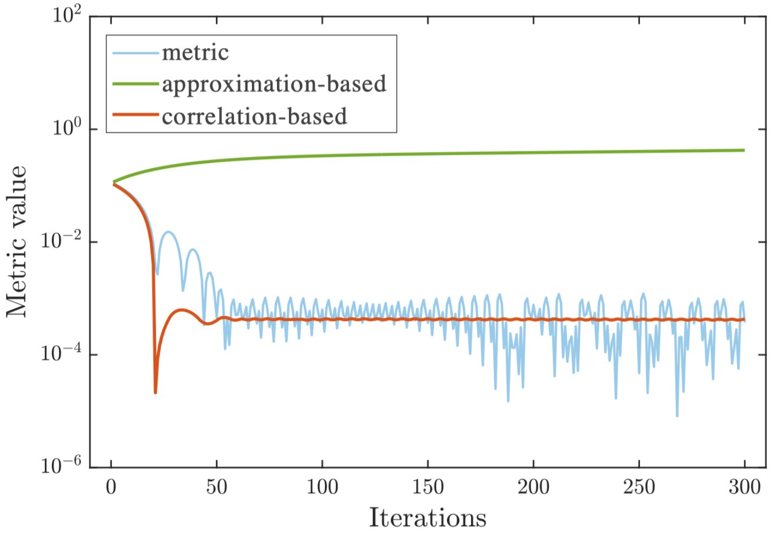
Relational Surrogate Loss Learning
Comparison to prior art

Learning Surrogate Losses [2]

- Train losses and models alternatively
- Train losses independently for each model
- Poor performance in our toy experiment

Relational Surrogate Loss Learning

- Train once for all the models of each task
- One learned loss generalizes to all the models
- Better performance



Convergent curves of toy experiment, lower is better.

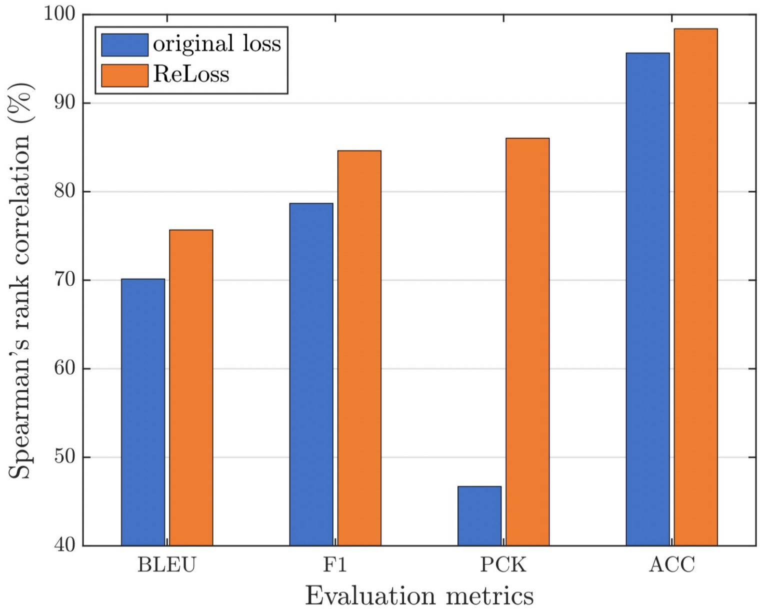
[2] Josif Grabocka, Randolf Scholz, and Lars Schmidt-Thieme. Learning surrogate losses. arXiv. preprint arXiv:1905.10108, 2019.

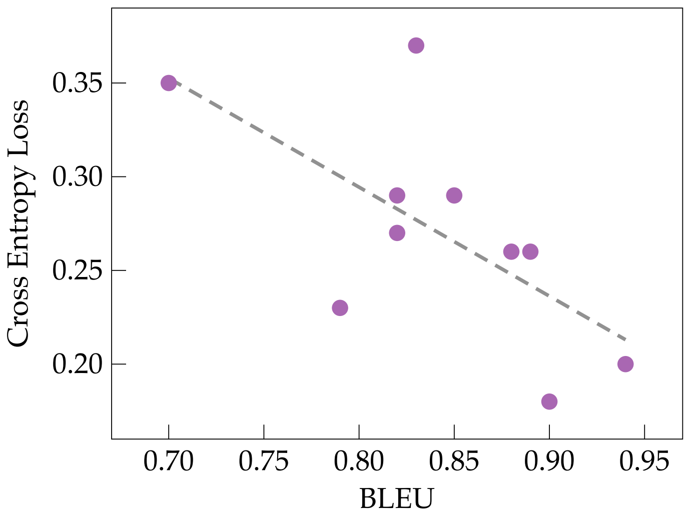
Relational Surrogate Loss Learning

Experiments

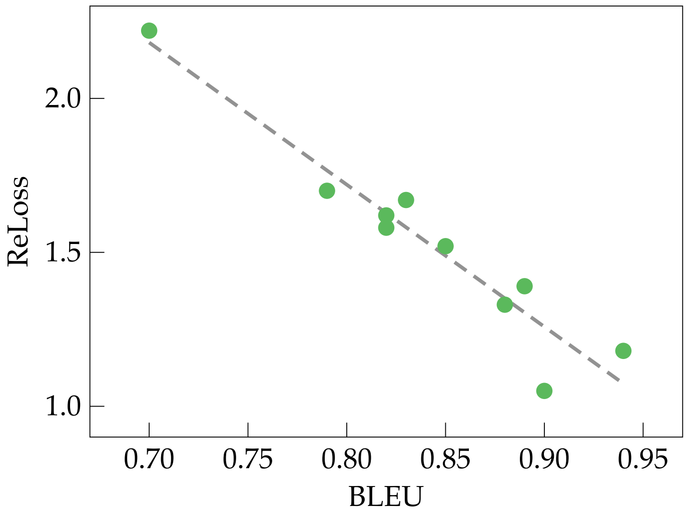
Better rank correlations compared to the original loss functions in

- Neural machine translation: BLEU
- Machine reading comprehension: F1
- Human pose estimation: PCK
- Image classification: ACC





BLEU vs. CE loss (Spearman: -0.58).



BLEU vs. ReLoss (Spearman: -0.90).

Relational Surrogate Loss Learning

Experiments

Better performance on both CV and NLP tasks:

Image Classification

Dataset	Model	CE		ReLoss	
		Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5 (%)
CIFAR-10	ResNet-56	94.32 ± 0.25	-	**94.57** ± 0.08	-
CIFAR-100	ResNet-56	73.61 ± 0.11	-	**74.15** ± 0.14	-
ImageNet	ResNet-50	76.5	93.0	**76.8**	93.0
	MobileNet V2	71.8	90.3	**72.2**	90.5

Neural Machine Translation

Model	Speed	Original loss		ReLoss on EN-RO		ReLoss on RO-EN	
		EN-RO	RO-EN	EN-RO	RO-EN	EN-RO	RO-EN
Transformer (Vaswani et al., 2017)	1.0×	32.88	33.94	-	-	-	-
NAT-Base (Gu et al., 2017)	15.6×	29.24	28.97	30.07 +0.83	29.68 +0.71	29.93 +0.69	29.61 +0.64
BoN-L_1(N=2)* (Shao et al., 2021)	15.6×	30.76	30.46	30.96 +0.20	30.74 +0.28	30.88 +0.12	30.78 +0.32

Human Pose Estimation *(outperforms SOTA)*

Method	Backbone	Input size	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR	PCK@0.05
validation set									
SimpleBaseline (Xiao et al., 2018)	ResNet-50	256 × 192	70.4	88.6	78.3	67.1	77.2	76.3	85.0
SimpleBaseline + ReLoss	ResNet-50	256 × 192	**71.9**	**89.9**	**80.0**	**68.0**	**77.9**	**77.3**	**86.1**
HRNet (Sun et al., 2019)	HRNet-W32	256 × 192	74.4	**90.5**	81.9	70.8	81.0	79.8	86.7
HRNet + ReLoss	HRNet-W32	256 × 192	**74.8**	**90.5**	**82.4**	**70.9**	**81.2**	**79.9**	**87.3**
test-dev set									
G-RMI (Papandreou et al., 2017)	ResNet-101	353 × 257	64.9	85.5	71.3	62.3	70.0	69.7	-
SimpleBaseline (Xiao et al., 2018)	ResNet-101	384 × 288	73.7	91.9	81.1	70.3	80.0	79.0	-
HRNet (Sun et al., 2019)	HRNet-W48	384 × 288	75.5	92.5	83.3	71.9	81.5	80.5	-
DARK (Zhang et al., 2020)	HRNet-W48	384 × 288	76.2	92.5	83.6	72.5	82.4	81.1	-
DARK + ReLoss	HRNet-W48	384 × 288	**76.4**	**92.7**	**83.7**	**72.7**	**82.5**	**81.3**	-

Machine Reading Comprehension *(outperforms SOTA)*

Method	ROUGE-L	BLEU-4	F1
dev set			
MacBERT-base (Cui et al., 2020)	51.4	50.3	53.9
MacBERT-base + ReLoss	**51.8**	**50.6**	**54.2**
MacBERT-large (Cui et al., 2020)	53.2	51.2	55.5
MacBERT-large + ReLoss	**53.6**	**51.4**	**55.9**
test set			
BiDAF† (Seo et al., 2016)	39.2	31.9	-
Wang et al. (2018)	44.2	41.0	-
MCR-Net-large (Peng et al., 2021)	50.8	49.2	-
Human Performance†	57.4	56.1	-
MacBERT-large + ReLoss	**64.9**	**61.8**	-

Relational Surrogate Loss Learning

Conclusion

- We study an interesting problem:

  Learning losses for non-differentiable evaluation metrics

- We use a simple method:

  Differentiable rank correlation to train better surrogate losses

- Potential applications:
  - New tasks with losses difficult to design
  - Existing tasks with losses align weak to evaluation metrics

Thank you!

The code is available at: https://github.com/hunto/ReLoss