Supplementary Materials of Efficient One Pass Self-distillation with Zipf's Label Smoothing

Jiajun Liang, Linze Li, Zhaodong Bing, Borui Zhao, Yao Tang, Bo Lin, and Haoqiang Fan

MEGVII Technology
{liangjiajun,lilinze,bingzhaodong,zhaoborui,tangyao02,linbo,fhq}@megvii.com

1 Explanation to empirical observation

We find that the Zipf's prior could help generate non-uniform supervision for non-target classes in a one-pass way. In this section, we provide a simple intuition to explain why Zipf's law should occur for predictions from multi-class classification.

We postulate that one main source of the non-zero network predictions is the inevitable non-orthogonality of the inter-class feature vectors as more and more classes are packed into the finite-dimensional feature space. In a simplified model, we assume that the decision vectors corresponding to each class are uniformly distributed on a high-dimensional unit sphere. Then for another random query vector on the sphere, their inner-products with it distribute in the shape of a Gaussian when the dimension is high enough.

We propose a toy experiment to verify that softmax Gaussian logits fit Zipf's law well. As shown in Algorithm 1, first, we sampled random vectors from multivariate normal distribution $\mathcal{N}(\mathbf{0}, I_{1000})$ as logits of different samples. Then logits for each sample are sorted and applied with softmax to get probabilities. At last, we average the sorted probabilities across all samples and plot the probabilities-

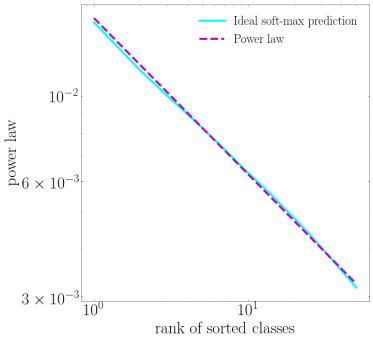


Fig. 1: softmax Gaussian logits fit Zipf's law well

rank relation for the top 32 categories in log-log space. It could be seen that a straight line pattern shows up in Figure 1.

Algorithm 1: simulation of ranking softmax Gaussian logits

```
# generate Gaussian logits,1000 samples, 1000 classes
logits = np.random.randn(1000, 1000)
# sort in class dimension
sorted_logits = np.sort(logits, axis=1)
# probability predictions by applying soft-max on the logits
sorted_preds = np.exp(sorted_logits) /
 np.sum(np.exp(sorted_logits),axis=1)[:,None]
# averaged across samples
mean_sorted_preds = np.mean(sorted_preds, axis=0)
# top 32 ranks considered
top32_sorted_preds = mean_sorted_preds[::-1][:32]
```

We also compare several most frequently-used distributions of long-tail shapes (Zipf's law, exponential and log-normal) to fit the empirical softmax scores, as shown in Table 1. All parameters of distributions are estimated by most-likelihood estimation. Common statistical test metrics such as R square are measured. Zipf's law outperforms the other two in all kinds of metrics.

Table 1: Statistical test of how well different distributions fit on empirical averaged predictions on INAT-21. The top 50 categories are considered. For tests such as Chisquare and Kolmogorov–Smirnov which heavily rely on the amount of the samples, we sample 10^5 instances from the empirical distribution. D is the Kolmogorov–Smirnov statistic and p is the p-value. Zipf's law outperforms the other two by all kinds of metrics.

Metric	Zipf's law	Exponential	Log-normal
R^2	0.99992	0.6768	0.9672
Kullback–Leibler divergence	0.0000667	0.315	0.0219
Jensen–Shannon divergence	0.0000167	0.063	0.00544
Chisquare	13.3	451677	4499
Kolmogorov–Smirnov	D=0.00278 p=0.42	D=0.265 p=0.0	D=0.0823 p=0.0

2 More Experiment Details and Discussion

2.1 Hyperparameters

Hyperparameters setting rules. Table 2 shows the detail of hyperparameters settings for different tasks. α controls the decay shape of Zipf's distribution,

Table 2: The detail of hyper parameters for different datasets.

Dataset	λ	α	dense layer	β
CIFAR100	0.1	1.0	2	0.1
TinyImageNet	1.0	1.0	2	0.5
ImageNet	0.1	1.0	1	/
INAT21	1.0	1.0	1	/

and is set to 1.0 in all tasks. λ controls the regularization strength, which is set to 0.1 for CIFAR100 and ImageNet, and 1.0 for TinyImageNet and NAT21. For datasets with large-resolution inputs, such as ImageNet and INAT, using the final dense feature maps would be sufficient, and no more dense layers are required. For low-resolution tasks such as CIFAR100 and TinyImageNet, we use one more dense layer to get enough votes for dense ranking. In this case, we need β to weigh the cross-entropy loss for learning the extra classifier. β is set to 0.1 and 0.5 respectively on CIFAR100 and TinyImageNet.

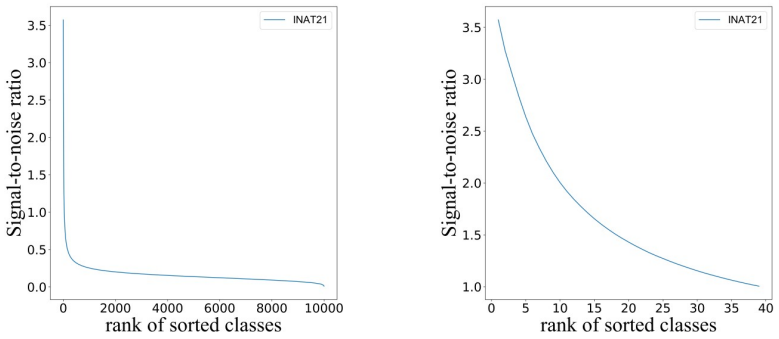
Hyperparameters ablation study. 1) α is not sensitive where $\alpha \in [0.5, 1.5]$. 2) λ is to control regularization strength and is positively correlated with the train/val acc gap. For tasks that are prone to overfitting(TinyImageNet and INAT whose train/val gap are 35% and 25%), λ is 1.0. For tasks that are less overfitting(ImageNet train/val gap is 4%), λ is 0.1. 3) β is optional and only recommended for small resolution tasks. It should be less than 0.5 to avoid shadow learning of deeper layers. See Table 3 for details.

Table 3: Ablation study of hyperparameters α,λ and β

α	0.1	0.5	1.0	1.5	2.0
CIFAR100	77.21±0.29	77.26±0.13	**77.38±0.32**	77.12±0.24	76.45±0.12
TinyImageNet	58.85±0.16	59.06±0.21	**59.25±0.20**	58.64±0.18	53.35±0.41
λ	0.01	0.1	0.5	1.0	1.5
CIFAR100	76.59±0.15	**77.38±0.32**	76.62±0.24	76.79±0.04	76.92±0.17
TinyImageNet	56.86±0.36	57.65±0.01	58.41±0.17	**59.25±0.20**	58.03±0.27
β(optional)	0.05	0.1	0.3	0.5	0.7
CIFAR100	77.24±0.22	**77.38±0.32**	76.75±0.31	76.39±0.16	76.49±0.24
TinyImageNet	58.71±0.17	58.77±0.18	59.48±0.31	**59.25±0.20**	59.00±0.12

2.2 SNR of Ranking

Ranking the classes accurately is a key factor to generate proper Zipf's law distribution for the sample. In the method section, we propose a finer ranking method named dense classification ranking which exploits spatial classification results from the last few feature maps. A consequence of this voting-based method is



(a) SNR-Rank relation across all classes

(b) SNR-Rank relation across top 40 classes

Fig. 2: SNR-Rank relation plot for trained ResNet-50 model in INAT21 dataset. It can be seen that only the top 40 rankings have SNR larger than 1, which makes it hard to give reliable ranks for tailing class on the fly.

that we have to clip the Zipf's values to a uniform one after a certain rank, as they would not receive sufficient votes to be distinguished individually. To illustrate that ranking only a few top non-target classes is sufficient, we study the signal-to-noise ratio of rankings. The signal and noise of specific rank r are calculated as the average and standard deviation among r-th probabilities from different sorted samples respectively. As shown in Figure.2, we plot the SNR-rank curve on the INAT21 dataset, only the top 40 out of 10000 ranks whose SNR is larger than one. It's a good trade-off to just give power-law decayed probabilities to the top-ranking class since the SNR of tailing ranks is too low to give reliable ranks.

2.3 More Zipf's Soft Label Examples

Figure 4 illustrates more results of the top-5 predictions of our proposed method compared with the baseline method. The top three rows are sampled from ImageNet while the bottom three rows are sampled from INAT21. It can be seen that: (1) There are several categories similar to the target class shown up in Zipf's soft labels, which provide meaningful label representations for the network to better grasp the concept of the target class. (2) Fine-grained categories of the target class emerge in Zipf's soft labels, which can provide similar "dark knowledge" as knowledge distillation.

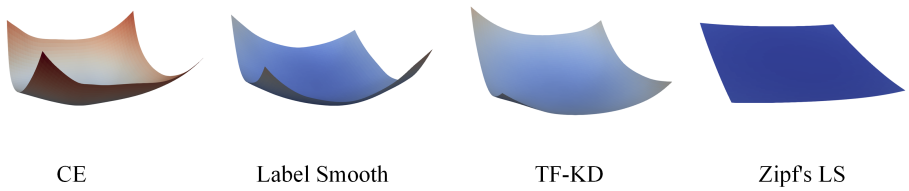


Fig. 3: Comparison of loss landscape with efficient teacher-free methods

2.4 Generalization: Performance on downstream tasks

To measure the power of generalization of Zipf's LS, we conducted the transfer learning task by fine-tuning ImageNet pre-trained models on MS-COCO, as shown in Table 4. Besides, we visualize loss landscapes [2] of several efficient teacher-free methods(see Fig 3), Zipf's LS achieves more flat convergence, which is a possible hint for better generalization [1].

Table 4: ImageNet pretrained ResNet50 for object detection

Method	Vanilla(CE)	TF-KD	PS-KD	Zipf's LS(Ours)
AP	36.4%	36.4%	36.5%	**36.6%**
AP@0.5	58.3%	56.7%	56.7%	**58.8%**

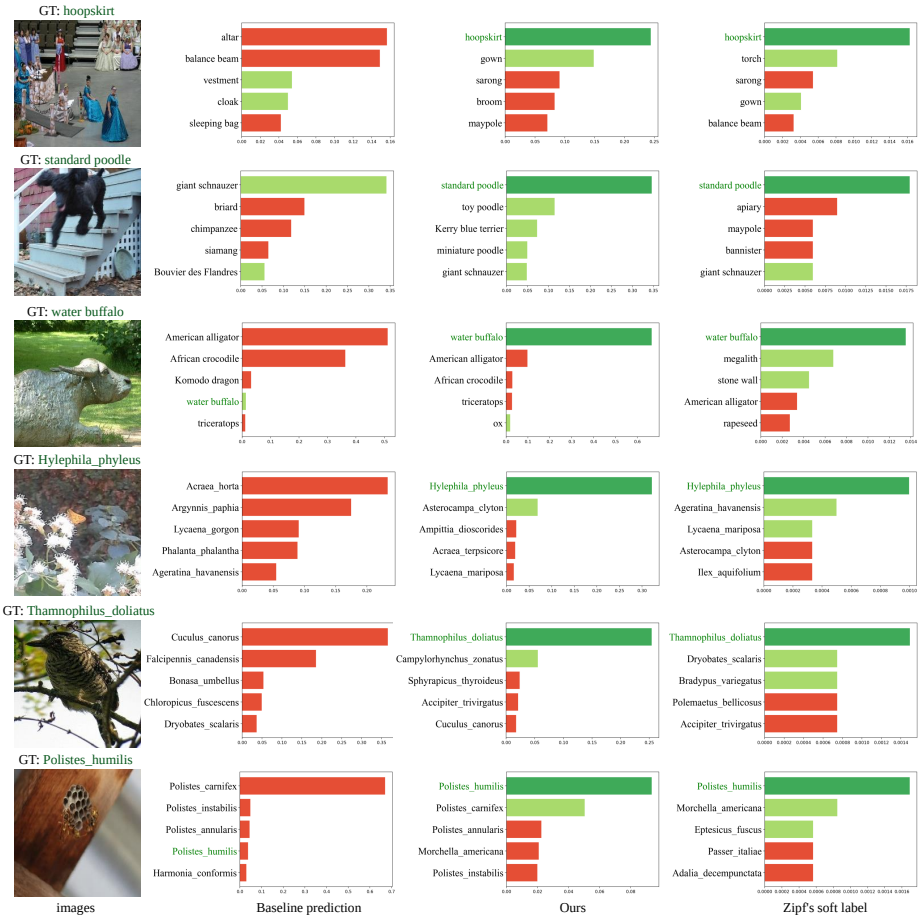


Fig. 4: Top-5 predictions visualization of our proposed method (Zipf's label smoothing) compared with the baseline method (cross-entropy). The dark green, light green and red denote ground-truth, similar and irrelevant categories respectively. "GT" denotes the ground truth label and thus the hard label of the baseline method. The baseline prediction is acquired under the supervision of the hard label and misclassified on the samples. Our method exploits target-relevant categories to better represent the image, and obtains better results.

References

1. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. UAI (2018)
2. Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T.: Visualizing the loss landscape of neural nets. In: NeurIPS (2018)