Improving Knowledge Distillation via Regularizing Feature Direction and Norm
– *Supplementary Material* –

Yuzhu Wang[1], Lechao Cheng[2,✉], Manni Duan[1], Yongheng Wang[1], Zunlei Feng[3], and Shu Kong[4,5,6]

[1]Zhejiang Lab [2]Hefei University of Technology [3]Zhejiang University
[4]University of Macau [5]Institute of Collaborative Innovation [6]Texas A&M University

Outline

As elaborated in the main paper, we proposed improve Knowledge Distillation (KD) by regularizing feature direction and norm. Our three major technical insights are (1) we take a novel perspective to improve KD by regularizing `student` *to produce features that are aligned with class-means features computed by the* `teacher` *and have sufficiently large norms, (2) we study multiple baseline methods to achieve such regularizations, and (3) we propose a novel and simple loss that simultaneously regularizes feature direction and norm, termed dino-loss. Experiments demonstrate that additionally adopting our dino-loss helps existing KD methods achieve better performance.*

In this supplementary document, We expand on the techniques of regularizing feature direction and norm, including implementation details, additional ablation studies and experiments.

Section A provides more implementation details for training and features aligning;

Section B presents additional ablation studies, including

- Sec B.1 studies the impact of hyper-parameters α and β in the KD++ loss;
- Sec B.2 compares KD++ and other methods in terms of computational complexity and the number of model parameters;
- Sec B.3 presents more visualization of embedding features;
- Sec B.4 studies the impact of magnitude of `teacher` feature norm;
- Sec B.5 experiments with larger teacher model and distills ViT to CNN;
- Sec B.6 analyze the impact of sampling strategy for class-means features;

Section C provides more experiment results and remarks, including

- Sec C.1 applies KD++ to non-classification tasks;
- Sec C.2 tries other L_p norms;
- Sec C.3 explains why one feature is projected and the other rotated in Fig. 3;

Section D summarizes potential limitations of this paper, which also points the way to future research directions.

A More Implementation Details

For fair comparisons, our implementation adheres to the previous methodologies outlined in [1, 7, 21, 27].

CIFAR-100 [9] contains 50k training images and 10k testing images. For each input image, 4 pixels are added as padding on each side, and a 32×32 cropping patch is randomly selected from the padded images or their horizontally flipped counterparts. We employ weight initialization as described in [4], training all `student` networks from scratch, while the teachers load the publicly available weights from [21]. The `student` networks are trained using a mini-batch size of 128 over 240 epochs (with a linear warmup for the first 20 epochs), employing SGD with a weight decay of 5e-4 and momentum of 0.9. We set the initial learning rate of 0.1 for ResNet [5] and WRN [26] backbones, and 0.02 for MobileNet [20] and ShuffleNet [14] backbones, decaying it with a factor of 10 at 150th, 180th, and 210th. The temperature is empirically set to 4.

ImageNet [19] comprises 1.28 million training images and 50,000 validation images spanning by 1,000 categories. We employ SGD with a mini-batch size of 512 for a total of 100 epochs (with a linear warmup for the first 5 epochs). The initial learning rate is set to 0.2 and is reduced by a factor of 10 every 30 epochs. Besides, the weight decay and momentum are set to 1e-4 and 0.9, respectively. The pre-trained weights for teachers come from PyTorch[1] and TIMM [23] for fair comparisons. The temperature for knowledge distillation is set to 1.

COCO 2017 [13] consists of 80 object categories with 118k training images and 5k validation images. We utilize Faster R-CNN [18] with FPN [12] as the feature extractor, and employ the dino-loss on the R-CNN head, wherein both `teacher` and `student` models adopt ResNet [5]. In addition, MobileNet-V2 [20] is used as a heterogeneous `student` model. All `student` models are trained with 1x scheduler, following Detectron2.[2]

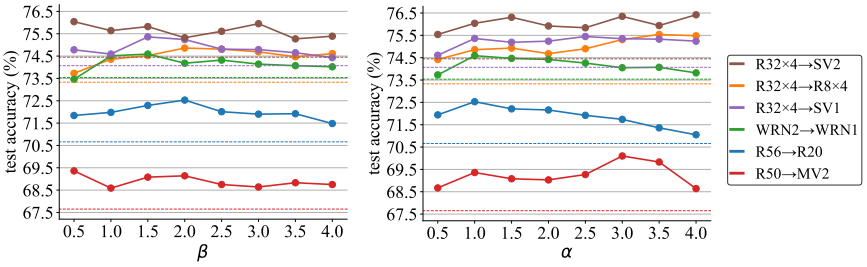
Our proposed dino-loss function regularizes the norm and direction of the `student` features at the penultimate layer before logits. The embedding features of the `student` and `teacher` models may have different dimensions. This can be addressed by learning a fully connected layer (followed by Batch Normalization) with the `student` to project its features to the same dimension as the `teacher`'s.

B Additional Ablation Studies

B.1 The sensitivity of hyper-parameters α and β

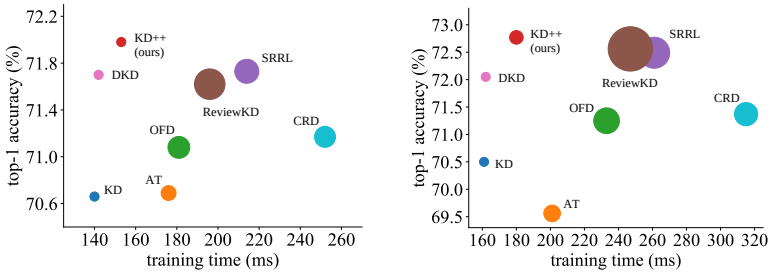
In the Eq. 8 , we introduce the KD++ loss function as $\mathcal{L} = \mathcal{L}_{ce} + \alpha\mathcal{L}_{kd} + \beta\mathcal{L}_{dino}$. For different `teacher-student` pairs, we perform grid search on α and β to get better results [1, 7, 27]. To substantiate the efficacy of the proposed dino-loss, we conduct extensive experiments aiming at probing the sensitivities of the hyperparameters α and β, as depicted in Fig. A1. The dashed lines illustrate the vanilla KD

[1] https://pytorch.org/vision/stable/models.html
[2] https://github.com/facebookresearch/detectron2



(a) sensitivity of β, (set $\alpha = 1$) **(b)** sensitivity of α, (select the best β)

Fig. A1: The sensitivity of hyper-parameters α and β. The dashed lines as baseline illustrate the performance based on vanilla KD loss. As α and β change, our proposed dino-loss can always achieve significant improvements compared to the baseline.



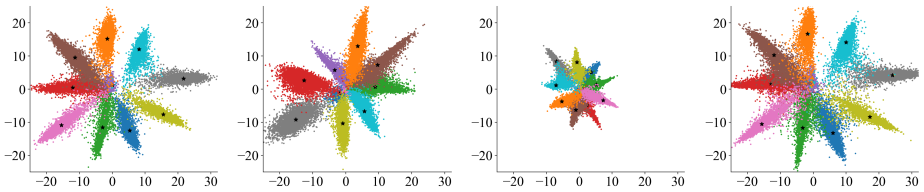
(a) ResNet-34→ResNet-18 **(b)** ResNet-50→MobileNet-V1

Fig. A2: Wall-clock time per training iteration *vs.* accuracy on the ImageNet validation set. (a): homogeneous architectures, (b): heterogeneous architectures. Enlarged circles correspond to a higher demand for parameters. KD++ achieves a better balance in terms of computational complexity, accuracy and number of parameters.

loss [6] (corresponding to specific setting, $\beta = 0$). As β ranges from 0.5 to 4.0, the solid line always surpasses the dashed line for the same color, indicating that our proposed dino-loss consistently surpasses KD-loss. Furthermore, in Fig. A1b, when the optimal β value is fixed, the distilled performance exhibits consistent enhancement compared to the baseline as α varies. These results compellingly attest to the overarching efficacy of the proposed dino-loss in our experiments, with the sensitivity of hyperparameters merely influencing the magnitude of improvement.

B.2 Complexity Comparisons

In this subsection, we compare the computational complexity, accuracy and number of parameters introduced between KD++ and other mainstream KD methods on the ImageNet dataset, as illustrated in Fig. A2. We measure the average time cost per batch iteration over the entire dataset as the horizontal axis



(a) `teacher` (Res56) **(b)** naive model (Res20) **(c)** `student` by KD **(d)** `student` by KD++

Fig. A3: Embedding features visualization on CIFAR-10. `Teacher` and `student` are ResNet-56 and ResNet-20, respectively. The same color belongs to the same category. $\star$ mean that class centers. Compared to KD, our method, KD++, achieves better feature alignment between `student` and `teacher`, and enables `student` to output features with larger norms.

and the Top-1 accuracy as the vertical axis. The varying sizes of circular markers representing different methods are proportional to the actual model parameter sizes. It is clear that our approach (KD++) delivers better performance with a small amount of time expense. It is important to highlight that in heterogeneous knowledge distillation tasks, there is typically a disparity in feature dimensions. Consequently, the inclusion of a bridging linear dimension transformation layer becomes imperative, attributing to the marginal increment in parameterization observed in our method, KD++, as compared to the classical KD approach.

B.3 More Visualization of Embedding Features

Although PCA [17] or t-SNE [15] have proven to be effective nonlinear dimensionality reduction techniques, we still adhere to the common practice of providing a more intuitive understanding. Therefore, we follow the approach of [22, 24] and introduce a 2-dimensional learnable feature output at the feature layer for visual analysis. We select the feature statistics of 10 classes from the `teacher` and `student` models on CIFAR-10 and visualize their 2D features, as shown in Fig. A3. Compared to KD, our method, KD++, achieves better feature alignment between `student` and `teacher`, and enables `student` to output features with larger norms.

B.4 Does the Magnitude of Teacher Norm Matter?

In the main paper, We found that allowing `student` to learn features with larger norms can benefit knowledge distillation. Therefore, a natural question arises: does increasing the `teacher` norm also contribute to improving KD ? To investigate this, we conduct experiments where we introduce a scaling factor, denoted as m, to the norm of the `teacher` in Eq. 7 as follows:

$$\mathcal{L}_{dino} = -\frac{1}{C}\sum_{k=1}^{C}\frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\frac{\mathbf{f}_i^s\cdot\mathbf{e}_k}{\max\{\|\mathbf{f}_i^s\|_2,\|\mathbf{f}_i^t\|_2\cdot(1+m)\}} \tag{1}$$

Table B1: Altering the norm of the `teacher` mode with a scaling factor m. Classification accuracy on the CIFAR-100 test set. The gray background indicates the default setting.

m	-0.5	-0.1	0.0	0.1	0.5	0.7	1.0	1.5	2.0
ResNet-56→ ResNet-20	71.57	72.19	**72.53**	71.76	71.86	71.64	71.79	71.74	71.92
ResNet-50→ MobileNet-V2	69.46	69.43	70.10	70.17	**70.23**	69.68	69.72	68.49	69.44

Table B2: Our method can benefit from larger `teacher` models. Methods are reported with top-1 accuracy (%) on the ImageNet validation set. With `teacher` capacity increasing, `student` models (trained with our dino-loss) achieve better classification results. Yet, previous methods do not necessarily obtain better results. All results are the average over 5 trials. We mark standard deviation using superscripts in green.

`student`	`teacher`	`student`	`teacher`	KD [6]	ReviewKD [1]	DKD [27]	KD++	ReviewKD++	DKD++
ResNet-18	ResNet-34	69.76	73.31	70.68$^{\pm 0.098}$	71.62$^{\pm 0.031}$	71.77$^{\pm 0.072}$	71.99$^{\pm 0.082}$	71.65$^{\pm 0.051}$	**72.08**$^{\pm 0.047}$
	ResNet-50		76.16	71.35$^{\pm 0.062}$	71.09$^{\pm 0.047}$	71.85$^{\pm 0.054}$	**72.49**$^{\pm 0.093}$	71.73$^{\pm 0.041}$	72.11$^{\pm 0.042}$
	ResNet-101		77.37	71.09$^{\pm 0.095}$	70.95$^{\pm 0.050}$	72.08$^{\pm 0.063}$	**72.54**$^{\pm 0.036}$	71.79$^{\pm 0.031}$	72.29$^{\pm 0.066}$
	ResNet-152		78.31	71.13$^{\pm 0.057}$	71.39$^{\pm 0.044}$	71.87$^{\pm 0.060}$	**72.59**$^{\pm 0.086}$	71.86$^{\pm 0.051}$	72.47$^{\pm 0.065}$
ResNet-18	ViT-S	69.76	74.64	71.32$^{\pm 0.061}$	n/a	71.21$^{\pm 0.068}$	**71.46**$^{\pm 0.032}$	n/a	71.33$^{\pm 0.043}$
	ViT-B		78.00	71.63$^{\pm 0.054}$	n/a	71.62$^{\pm 0.071}$	**71.84**$^{\pm 0.066}$	n/a	71.69$^{\pm 0.075}$

Interestingly, our experimental results (Table B1) indicate that in the context of homogeneous knowledge distillation, altering the norm of the `teacher`, whether increasing or decreasing it, does not lead to better improvement in `student` performance compared to maintaining the original norm of the `teacher`. However, in the case of heterogeneous knowledge distillation, there may be benefits in appropriately increasing the norm of the `teacher` features. It is worth noting that since this experiment has not been tested on a large-scale dataset, we cannot definitively conclude whether a larger `teacher` norm will always result in improvements. Nonetheless, this presents a promising direction for future exploration, where joint constraints on the norm size and direction can be applied to both `teacher` and `student`.

B.5 Experiments with larger teacher model

Fig. 5 clearly shows that our method can benefit from larger `teacher` models. We report the mean top-1 accuracy on the validation with standard deviation over five runs, and the results of distillation from ViT [3] to ResNet in Table B2. KD++ consistently outperforms the competitions. Nonetheless, owing to the architectural differences, specifically the contrasting characteristics of global attention in Transformer and local receptive fields in Convolution, the benefits are not as conspicuous as in cases with homogeneous architectures.

B.6 The Sample Selection Strategy for Class Mean

For small-scale datasets such as CIFAR, we compute the class-mean features in the entire training set, as models could achieve close to 100% accuracy on the training set. However, for large-scale datasets like ImageNet, the models

exhibit lower accuracy on the training set (e.g., 73.31% for ResNet-34). In such cases, using all training samples to evaluate class centers would inevitably impact the distribution of each class center. We investigate two methods for computing class centers on ImageNet: (1) utilizing all samples and (2) only considering the correctly predicted samples by the `teacher` model. It is important to note that all samples are derived from the training set. The `teacher` and `student` models are ResNet-34 and ResNet-18. We found that the result (72.01%) by only the correctly predicted samples by the `teacher` slightly outperforms using all samples (71.98%). This confirms the existence of this issue in large-scale datasets; however, the impact is insignificant, we default to using all samples for computing class-mean features.

C Additional Experiments Results

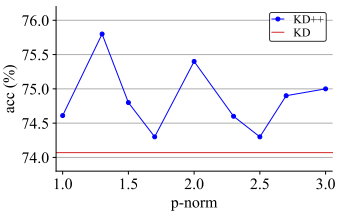
C.1 Apply KD++ to non-classification tasks

We test our dino-loss atop of CLIP-KD [25] that studies distilling CLIP models, which are implemented as transformers and state-of-the-art for vision-language learning. In particular, we study KD through the lens of retrieval, which does not work on categorical labels. While our method requires computing discrete features (i.e., class-mean features in classification), we circumvent the class-mean feature computing by first constructing a pool of object concepts parsed from the text corpus, and then using the `teacher` to create *pseudo labels* for input images. The `teacher-student` pairs, text encoder and training details follow CLIP-KD. Results in Table B3 demonstrate that our method generalizes well to more advanced network architectures and tasks beyond classification tasks.

Table B3: We report R@1 of Image-to-Text (I2T) and Text-to-Image (T2I) retrieval, and zero-shot accuracy on ImageNet-1K as a by-product.

Methods	CC3M val		COCO		IN-1K
	I2T	T2I	I2T	T2I	top-1
T: ViT-B/16	36.99	40.16	39.54	24.98	24.66
S: ViT-T/16	30.55	33.34	33.25	20.70	20.30
CLIP-KD [25]	34.23	37.13	36.85	22.52	22.16
CLIP-KD + ours	**38.39**	**38.03**	**23.28**	**22.91**	**35.14**

Fig. A4: Our dino-loss is robust to the choice of p.



C.2 Try other L_p norms

It's interesting to try other L_p norms, so we plot the curve below between accuracy *vs.* p in our dino-loss by distilling R32×4 `teacher` to SV1 `student` on CIFAR-100 in Fig. A4, showing that our dino-loss is robust to the choice of p.

C.3 Remarks on feature rotation and projection

For each training example, we project its `student` feature but rotate its `teacher` feature to the class-mean direction (see Fig. 3). As shown in Fig. 2, different

training examples have different feature norms computed by a trained model. We hypothesize that, given a data example, the norm of its `teacher` feature can better guide how large the feature norm is by the `student`, whereas the class-mean guides the `student` feature direction. (1) For `teacher`, we rotate rather than project its features to preserve its norm; (2) For `student`, we project its features to class-mean for directional regularization; (3) Importantly, our dino-loss imposes instance-aware regularization during training. This may explain why our dino-loss performs the best.

D Limitations

Our work has some visible limitations, e.g., we do not study how to distill large pretrained foundation models (e.g., SAM [8] and GPT [16]). Moreover, as our methods are orthogonal to existing ones such as search-based methods [2, 10, 11], it is worth studying how incorporating our techniques improve search-based KD methods.

References

1. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling knowledge via knowledge review. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5008–5017 (2021)
2. Dong, P., Li, L., Wei, Z.: Diswot: Student architecture search for distillation without training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11898–11908 (2023)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
7. Huang, T., You, S., Wang, F., Qian, C., Xu, C.: Knowledge distillation from a stronger teacher. arXiv preprint arXiv:2205.10536 (2022)
8. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything (2023)
9. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
10. Li, L., Dong, P., Li, A., Wei, Z., Ya, Y.: Kd-zero: Evolving knowledge distiller for any teacher-student pairs. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)

11. Li, L., Dong, P., Wei, Z., Yang, Y.: Automated knowledge distillation via monte carlo tree search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17413–17424 (October 2023)
12. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
14. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
15. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008)
16. OpenAI: Gpt-4 technical report (2023)
17. Pearson, K.: Principal component analysis. Mathematical Proceedings of the Royal Society of London **PCA**(97), 405–413 (1901). https://doi.org/10.1098/rspa.1901.0010
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015)
20. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
21. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. arXiv preprint arXiv:1910.10699 (2019)
22. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. pp. 499–515. Springer (2016)
23. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861
24. Xu, R., Li, G., Yang, J., Lin, L.: Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1426–1435 (2019)
25. Yang, C., An, Z., Huang, L., Bi, J., Yu, X., Yang, H., Diao, B., Xu, Y.: Clip-kd: An empirical study of clip model distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
26. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
27. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11953–11962 (2022)